5 Series MSO (MSO54, MSO56, MSO58, MSO58LP)
6 Series MSO (MSO64)
Programmer Manual
5 Series MSO (MSO54, MSO56, MSO58, MSO58LP)
6 Series MSO (MSO64)

Programmer Manual
# Table of Contents

Preface .............................................................................................................. iii

Getting Started .................................................................................................... 1-1

Setting Up Remote Communications Hardware ......................................................... 1-1
  Ethernet .................................................................................................. 1-1
  USB ....................................................................................................... 1-2

Setting Up Remote Communications Software .......................................................... 1-3
  Using VISA .............................................................................................. 1-3
  Using the LXI Web Page and e*Scope ............................................................... 1-4
  Using a Socket Server .................................................................................. 1-5

Documentation .............................................................................................. 1-7
  Dynamic programmatic interface .......................................................................... 1-7
  Implicit activation ........................................................................................... 1-7
  Creating, deleting, and listing dynamic instances ........................................................ 1-8

Command Syntax ................................................................................................. 2-1
  Command and Query Structure ............................................................................ 2-1
  Clearing the oscilloscope ................................................................................... 2-3
  Command Entry .............................................................................................. 2-3
  Constructed Mnemonics .................................................................................... 2-5
  Argument Types .............................................................................................. 2-7

Command groups............................................................................................... 2-11
  Acquisition command group ............................................................................. 2-11
  AFG Command Group .................................................................................... 2-12
  Alias command group ..................................................................................... 2-13
  Bus command group ....................................................................................... 2-14
  Calibration command group .............................................................................. 2-19
  Cursor command group ................................................................................... 2-20
  Digital command group .................................................................................. 2-26
  Display control command group ......................................................................... 2-28
  DVM Command Group ................................................................................... 2-31
  Ethernet Command Group ................................................................................ 2-31
  File system command group .............................................................................. 2-33
  Horizontal command group ............................................................................... 2-35
  Mask command group ..................................................................................... 2-37
  Math command group ..................................................................................... 2-38
  Measurement command group .......................................................................... 2-40
  Miscellaneous command group .......................................................................... 2-54
  Plot command group ...................................................................................... 2-56
  Power command group .................................................................................... 2-56
  Save and Recall command group ...................................................................... 2-69
<table>
<thead>
<tr>
<th>Table of Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save On command Group</td>
<td>2-70</td>
</tr>
<tr>
<td>Search and Mark command group</td>
<td>2-71</td>
</tr>
<tr>
<td>Self Test command group</td>
<td>2-86</td>
</tr>
<tr>
<td>Status and Error command group</td>
<td>2-87</td>
</tr>
<tr>
<td>Trigger command group</td>
<td>2-88</td>
</tr>
<tr>
<td>Vertical command group</td>
<td>2-99</td>
</tr>
<tr>
<td>Waveform Transfer command group</td>
<td>2-102</td>
</tr>
<tr>
<td>Zoom command group</td>
<td>2-106</td>
</tr>
<tr>
<td>Commands listed in alphabetical order</td>
<td>2-107</td>
</tr>
<tr>
<td>Status and Events</td>
<td>3-1</td>
</tr>
<tr>
<td>Registers</td>
<td>3-1</td>
</tr>
<tr>
<td>Queues</td>
<td>3-4</td>
</tr>
<tr>
<td>Event Handling Sequence</td>
<td>3-5</td>
</tr>
<tr>
<td>Synchronization Methods</td>
<td>3-7</td>
</tr>
<tr>
<td>Appendix A: Character Set</td>
<td>A-1</td>
</tr>
<tr>
<td>Appendix B: Reserved Words</td>
<td>B-1</td>
</tr>
<tr>
<td>Appendix C: Factory Defaults</td>
<td>C-1</td>
</tr>
<tr>
<td>Default Setup</td>
<td>C-1</td>
</tr>
<tr>
<td>Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples</td>
<td>D-1</td>
</tr>
<tr>
<td>Example 1: Analog Waveform (Channels 1–4)</td>
<td>D-1</td>
</tr>
<tr>
<td>Example 2: Digital Waveform</td>
<td>D-3</td>
</tr>
<tr>
<td>Example 3: Digital with 4 Bytes Per Point and Zoom Off</td>
<td>D-5</td>
</tr>
<tr>
<td>Example 4: Digital with 8 Bytes Per Point and Zoom Off</td>
<td>D-8</td>
</tr>
<tr>
<td>Example 5: Digital with 4 Bytes Per Point and Zoom On</td>
<td>D-10</td>
</tr>
<tr>
<td>Example 6: Digital with 8 Bytes Per Point and MagniVu On</td>
<td>D-13</td>
</tr>
<tr>
<td>Example 7: RF Frequency Domain Waveform</td>
<td>D-15</td>
</tr>
<tr>
<td>Appendix E: Search and Trigger Command Sequence Examples</td>
<td>E-1</td>
</tr>
<tr>
<td>Example 1: Single Threshold Edge Search</td>
<td>E-1</td>
</tr>
<tr>
<td>Example 2: Single Threshold Edge Trigger</td>
<td>E-2</td>
</tr>
<tr>
<td>Example 3: Dual Threshold Runt Search</td>
<td>E-2</td>
</tr>
<tr>
<td>Example 4: Single Threshold Logic Search on Three Waveforms</td>
<td>E-3</td>
</tr>
<tr>
<td>Glossary</td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td></td>
</tr>
</tbody>
</table>
Preface

This programmer guide provides you with the information required to use Programmable Interface commands to remotely control your instrument.

The programmer manual is divided into the following major topics:

- **Getting started.** This topic introduces you to the programming information and provides basic information about setting up your instrument for remote control.

- **Command syntax.** This topic provides an overview of the command syntax that you use to communicate with the instrument and other general information about commands, such as how commands and queries are constructed, how to enter commands, constructed mnemonics, and argument types.

- **Command groups.** This topic contains all the commands listed in functional groups. Each group consists of an overview of the commands in that group and a table that lists all the commands and queries for that group. You can click a command in the listing to display a detailed description of the command.

- **Commands listed in alphabetical order.** This topic contains all commands listed in alphabetical order. Command details, syntax, and examples are provided.

- **Status and events.** This topic discusses the status and event reporting system for the programming interfaces. This system informs you of certain significant events that occur within the instrument. Topics that are discussed include registers, queues, event handling sequences, synchronization methods, and messages that the instrument might return, including error messages.

- **Appendices.** These topics contain miscellaneous information, such as a list of reserved words, a table of the factory initialization (default) settings, and interface specifications that can be helpful when using commands to remotely control the instrument.
Getting Started

This manual explains the use of commands to remotely control your oscilloscope. With this information, you can write computer programs to perform functions, such as setting the front-panel controls, taking measurements, performing statistical calculations, and exporting data for use in other programs.

Familiarity with the User Manual for your oscilloscope is assumed. You can download the User Manual from the Tektronix website at www.tek.com.

NOTE. Most examples in this document assume that both HEADER and VERBose are set to ON.

Setting Up Remote Communications Hardware

You can remotely control communications between your oscilloscope and a PC via Ethernet or USB cables.

Ethernet

If you are using Ethernet, start by connecting an appropriate Ethernet cable to the Ethernet port (RJ-45 connector) on the rear panel of your oscilloscope. This connects the oscilloscope to a 10BASE-T/100BASE-T/1000BASE-T local area network.
To change the Ethernet settings on your oscilloscope, do the following:

1. Select the **Utility** drop-down menu.
2. Select the **I/O** menu.
3. Select the **LAN** panel.
4. In the menu, if you are on a DHCP Ethernet network that supplies the IP address automatically by a DHCP, tap **Auto**.
5. In the menu, if you want to supply your own network settings, tap **Manual** to set a hard coded TCP/IP address.

### USB

If you are using USB, start by connecting the appropriate USB cable to the USB 3.0 super-speed (SS) Device port on the rear panel of your oscilloscope. This port requires that the cable connected from the port to the host computer meets the USB 3.0 specification for super speed connections. Typically, such cables should be 3 feet or shorter in length, but this is determined by the quality of the cable and, with higher quality cables, this length can be extended. (It is also dependent upon the drive capability of the host USB port to which the instrument is connected.) The use of high quality short cables is recommended to avoid USB connection problems.

With USB, the system automatically configures itself. To verify that the USB is enabled:

1. Select the **Utility** drop-down menu.
2. Select the **I/O** menu.
3. Touch **USB Device Port** to open the USB Device Port configuration menu.
4. If USB is disabled, tap **USB Device Port** to enable the USB Device port.
After connection, the host, with appropriate software, can list the oscilloscope as a USB device with the following parameters: (See Table 1-1.)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer ID</td>
<td>0x0699 (decimal 1689)</td>
</tr>
<tr>
<td>Product ID</td>
<td>0x0522</td>
</tr>
<tr>
<td></td>
<td>(You can send the USBTMC:PRODUCTID:HEXadecimal? query to read the value)</td>
</tr>
<tr>
<td>Serial number</td>
<td>Serial number</td>
</tr>
<tr>
<td>Manufacturer description</td>
<td>“Tektronix”</td>
</tr>
<tr>
<td>Interface description</td>
<td>“USBTMC-USB488”</td>
</tr>
</tbody>
</table>

Setting Up Remote Communications Software

Connect your oscilloscope directly to a computer to let the PC analyze your data, collect screen images, or to control the oscilloscope using a program of your own creation. Three ways to connect your oscilloscope to a computer are through the VISA drivers, the e*Scope Web-enabled tools, or via a socket server.

Using VISA

VISA lets you use your MS-Windows computer to acquire data from your oscilloscope for use in an analysis package that runs on your PC, such as Microsoft Excel, National Instruments LabVIEW, Tektronix OpenChoice Desktop software, or your own custom software. You can use a common communications connection, such as USB or Ethernet, to connect the computer to the oscilloscope.

To set up VISA communications between your oscilloscope and a computer:
1. Load the VISA drivers on your computer. Also, load your application, such as OpenChoice Desktop. You will find the drivers and OpenChoice Desktop software on the appropriate CD that comes with your oscilloscope or at the Tektronix software finder Web page (www.tektronix.com/downloads).

2. Connect the oscilloscope to your computer with the appropriate USB or Ethernet cable. Cycle the power on the oscilloscope.

3. Select the Utility drop-down menu.

4. Select I/O menu.

5. If you are using USB, the system sets itself up automatically for you, if USB is enabled. Check USB DEVICE PORT panel to be sure that USB is enabled. If it is not enabled, toggle the On/Off button to On.

6. To use Ethernet, select the LAN panel. Use the controls to adjust your network settings, as needed. For more information, see the e*Scope setup information below.

7. If you want to change socket server parameters, select the Socket Server panel and enter new values through the resulting panel controls.

8. Run your application software on your computer.

Quick Tips

- The USB 3.0 super-speed (SS) device port is the correct USB port for computer connectivity. Use the rear- and front-panel USB 2.0 or 3.0 host ports to connect your oscilloscope to USB flash drives, hard drives, keyboards or mice. Use the USB Device port to connect your oscilloscope to a PC.

- There are both USB 2.0 and 3.0 host ports on the instrument. The device port is USB 3.0. Printers are not supported on host ports. PictBridge printers are not supported on the device port.

Using the LXI Web Page and e*Scope

With e*Scope, you can access your Internet-connected instrument from a web browser.

NOTE. This procedure presumes that you have connected the oscilloscope to a network and configures the oscilloscope to work on your network. See the oscilloscope Installation and Safety instructions, or the embedded Help on the oscilloscope, for instructions on how to configure the oscilloscope for network use.

To set up e*Scope communications between your oscilloscope and a web browser running on a remote computer:
1. Connect the oscilloscope to your computer network with an appropriate Ethernet cable.
2. Select the Utility drop-down menu.
3. Select the I/O menu.
4. Select the LAN panel.
5. At the top left of the panel, there is an indicator light which turns red if the device detects a fault.
6. Tap Test Connection to check if your oscilloscope can find an attached network.
7. Record the IP address shown on your oscilloscope.
8. Start a Web browser on a PC that has access to the same network to which the oscilloscope is attached.
9. Enter the IP address of the oscilloscope in the browser address line. You should then see the LXI Welcome page on your Web browser on your computer screen.
10. For e*Scope, click the Instrument Control (e*Scope) link on the left side of the LXI Welcome page. You should then see a new tab (or window) open in your browser, showing the oscilloscope screen. You can use the PC mouse to interact with the e*Scope screen and controls.

Using a Socket Server

A socket server provides two-way communication over an Internet Protocol-based computer network. You can use your oscilloscope’s socket server feature to let your oscilloscope talk to a remote-terminal device or computer.

To set up and use a socket server between your oscilloscope and a remote terminal or computer:
1. Connect the oscilloscope to your computer network with an appropriate Ethernet cable.
2. Select the Utility drop-down menu.
3. Select the I/O menu.
4. Tap Socket Server.
5. On the resulting Socket Server panel, tap the top entry to toggle the Socket Server On.
6. Choose whether the protocol should be None or Terminal. A communication session run by a human at a keyboard typically uses a terminal protocol. An automated session might handle its own communications without using such a protocol.
7. If required, change the port number by rotating multipurpose knob a.
8. If required, press OK to set the new port number.

9. After setting up the socket server parameters, you can now have the computer “talk” to the oscilloscope. If you are running an MS Windows PC, you could run its default client with its command-like interface. One way to do this is by typing “Telnet” in the Run window. The Telnet window will open on the PC.

**NOTE.** On MS Windows 7, you must first enable Telnet in order for it to work.

10. Start a terminal session between your computer and your oscilloscope by typing in an open command with the oscilloscope LAN address and port #. You can obtain the LAN address by pushing the LAN panel to view the resulting LAN setting panel. You can obtain the port # by tapping the Socket Server panel and viewing the Port item.

For example, if the oscilloscope IP address was 123.45.67.89 and the port # was the default of 4000, you could open a session by writing into the MS Windows Telnet screen:

```
open 123.45.67.89 4000
```

The oscilloscope will send a help screen to the computer when it is done connecting.

11. You can now type in a standard query, as found in the programmer manual, such as *idn? 

The Telnet session window will respond by displaying a character string describing your instrument. You can type in more queries and view more results on this Telnet session window. You can find the syntax for relevant queries and related status codes in other sections of this manual.

**NOTE.** Do not use the computer’s backspace key during an MS Windows’ Telnet session with the oscilloscope.

**Socket Server Terminal Protocol Mode Commands.** Following are Tektronix Instrument Control Terminal Session Control commands:

- `/t <timeout>`: set the response timeout in milliseconds.
- `/d`: send device clear to the instrument.
- `/r`: read response from instrument.
- `/h`: print this usage info.
Dynamic programmatic interface

This programmatic interface is dynamic. This means the instrument will not recognize certain commands until the objects referenced by those commands actually exist. For example, commands related to measurements are not recognized until measurements are added. Therefore, the response to a *LRN? query will not normally include the instrument's complete command set.

The following command groups are not available when the instrument is in its default state:

- Measurement
- Math
- Bus
- Search and Mark
- Plot

Adding an instance of one of those components will cause all commands related to that component to be recognized. For example, sending the `MEASUrement:ADDNew` command adds a measurement at which point the measurement commands will be recognized. Conversely, once all instances of a component have been deleted, the commands related to that component will no longer be recognized.

Implicit activation

When you send a command or query related to a dynamic object (such as Math1, Bus3, or Meas2) to the instrument and that instance does not yet exist, the instrument:

- creates a default instance with the name you specified
- adds all relevant commands and queries to the set of recognized commands
- responds to the command or query

Example 1

*RST followed by a *LRN? will not return any MATH<x> commands because in the default state, the instrument does not have any math waveforms. However querying :MATH:MATH1:DEFinition? will add MATH1 with the default math
expression Ch1 - Ch2. Then the query will return the expected result. Note that if Ch1 or Ch2 is not active, they will be activated as part of this action. A *LRN issued after this will return MATH commands in addition to other available commands.

**Example 2**

*RST followed by :MEASUrement:MEAS3:TYPE? creates a measurement named MEAS3 and return it's type. Since the default type is Period, you will get Period as the response. A *LRN issued after this will return all MEASUrement:MEAS3 commands in addition to other available commands.

Not all commands in these groups start implicit activation. ADDNew, DELete, and LIST commands do not result in implicit activation.

**Creating, deleting, and listing dynamic instances**

You can create a new default instance of a dynamic feature by using the ADDNew command. For example, :MEASUrement:ADDNew "Meas1" will create a new measurement named Meas1. Meas1 will be a Period measurement since the default type for measurements is Period. Note that you can change Meas1 type to any other supported type using :MEASUrement:MEAS1:TYPE command.

The DELete command deletes the named dynamic instance. For example, :BUS:DELete "B2" will delete a bus named B2 if it exists. You can delete only one instance at a time. The LIST command returns a list of all dynamic instances currently in existence. For example, if you have added three measurements named Meas1, Meas2 and Meas3, :MEASUrement:LIST? returns MEAS1, MEAS2, MEAS3.
You can control the operations and functions of the oscilloscope through the Ethernet port or the USB 3.0 super speed (SS) device port using commands and queries. The related topics listed below describe the syntax of these commands and queries. The topics also describe the conventions that the oscilloscope uses to process them. See the Command Groups topic in the table of contents for a listing of the commands by command group, or use the index to locate a specific command.

**Backus-Naur Form Notation**

This documentation describes the commands and queries using Backus-Naur Form (BNF) notation. Refer to the following table for the symbols that are used.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&gt;</td>
<td>Defined element</td>
</tr>
<tr>
<td>=</td>
<td>Is defined as</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>{ }</td>
<td>Group; one element is required</td>
</tr>
<tr>
<td>[ ]</td>
<td>Optional; can be omitted</td>
</tr>
<tr>
<td>...</td>
<td>Previous element(s) may be repeated</td>
</tr>
</tbody>
</table>

**Command and Query Structure**

Commands consist of set commands and query commands (usually called commands and queries). Commands modify oscilloscope settings or tell the oscilloscope to perform a specific action. Queries cause the oscilloscope to return data and status information.

Most commands have both a set form and a query form. The query form of the command differs from the set form by its question mark at the end. For example, the set command `ACQuire:MODe` has a query form `ACQuire:MODe?`. Not all commands have both a set and a query form. Some commands have set only and some have query only.

**Messages**

A command message is a command or query name followed by any information the oscilloscope needs to execute the command or query. Command messages may contain five element types, defined in the following table.
Table 2-2: Command Message Elements

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Header&gt;</td>
<td>This is the basic command name. If the header ends with a question mark, the command is a query. The header may begin with a colon (:) character. If the command is concatenated with other commands, the beginning colon is required. Never use the beginning colon with command headers beginning with a star (*).</td>
</tr>
<tr>
<td>&lt;Mnemonic&gt;</td>
<td>This is a header subfunction. Some command headers have only one mnemonic. If a command header has multiple mnemonics, a colon (:) character always separates them from each other.</td>
</tr>
<tr>
<td>&lt;Argument&gt;</td>
<td>This is a quantity, quality, restriction, or limit associated with the header. Some commands have no arguments while others have multiple arguments. A &lt;space&gt; separates arguments from the header. A &lt;comma&gt; separates arguments from each other.</td>
</tr>
<tr>
<td>&lt;Comma&gt;</td>
<td>A single comma is used between arguments of multiple-argument commands. Optionally, there may be white space characters before and after the comma.</td>
</tr>
<tr>
<td>&lt;Space&gt;</td>
<td>A white space character is used between a command header and the related argument. Optionally, a white space may consist of multiple white space characters.</td>
</tr>
</tbody>
</table>

Commands

Commands cause the oscilloscope to perform a specific function or change one of the settings. Commands have the structure:

[:]<Header>[<Space><Argument>[<Comma> <Argument>]]...

A command header consists of one or more mnemonics arranged in a hierarchical or tree structure. The first mnemonic is the base or root of the tree and each subsequent mnemonic is a level or branch off the previous one. Commands at a higher level in the tree may affect those at a lower level. The leading colon (:) always returns you to the base of the command tree.
Queries

Queries cause the oscilloscope to return status or setting information. Queries have the structure:

- [:]<Header>
- [:]<Header>[<Space><Argument> [Comma]<Argument>]

You can specify a query command at any level within the command tree unless otherwise noted. These branch queries return information about all the mnemonics below the specified branch or level.

Headers

Use the HEADer command to control whether the oscilloscope returns headers as part of the query response. If header is on, the query response returns command headers, then formats itself as a valid set command. When header is off, the response includes only the values. This may make it easier to parse and extract the information from the response. The table below shows the difference in responses.

<table>
<thead>
<tr>
<th>Query</th>
<th>Header Off</th>
<th>Header On</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME?</td>
<td>&quot;14:30:00&quot;</td>
<td>:TIME &quot;14:30:00&quot;</td>
</tr>
<tr>
<td>ACQuire:NUMAVg?</td>
<td>100</td>
<td>:ACQUIRE:NUMAVG 100</td>
</tr>
</tbody>
</table>

Clearing the oscilloscope

You can clear the Output Queue and reset the oscilloscope to accept a new command or query by using the selected Device Clear (DCL) function.

Command Entry

The following rules apply when entering commands:

- You can enter commands in upper or lower case.
- You can precede any command with white space characters. White space characters include any combination of the ASCII control characters 00 through 09 and 0B through 20 hexadecimal (0 through 9 and 11 through 32 decimal).
- The oscilloscope ignores commands consisting of any combination of white space characters and line feeds.
Abbreviating

You can abbreviate many oscilloscope commands. Each command in this documentation shows the minimum acceptable abbreviations in capitals. For example, you can enter the command `ACQuire:NUMAvg` simply as `ACQ:NUMA` or `acq:numa`.

Abbreviation rules may change over time as new oscilloscope models are introduced. Thus, for the most robust code, use the full spelling.

If you use the `HEADER` command to have command headers included as part of query responses, you can further control whether the returned headers are abbreviated or are full-length with the `VERBose` command.

Concatenating

You can concatenate any combination of set commands and queries using a semicolon (;). The oscilloscope executes concatenated commands in the order received.

When concatenating commands and queries, you must follow these rules:

1. Separate completely different headers by a semicolon and by the beginning colon on all commands except the first one. For example, the commands `TRIGger:MODe NORMal` and `ACQuire:NUMAVg 8`, can be concatenated into the following single command:

   `TRIGger:MODe NORMal;:ACQuire:NUMAVg 8`

2. If concatenated commands have headers that differ by only the last mnemonic, you can abbreviate the second command and eliminate the beginning colon. For example, you can concatenate the commands `ACQuire:MODe ENvelope` and `ACQuire:NUMAVg 8` into a single command:

   `ACQuire:MODe ENvelope; NUMAVg 8`

   The longer version works equally well:

   `ACQuire:MODe ENvelope;:ACQuire:NUMAVg 8`

3. Never precede a star (*) command with a colon:

   `ACQuire:STATE 1;*OPC`

   Any commands that follow will be processed as if the star command was not there so the commands, `ACQuire:MODe ENvelope;*OPC;NUMAVg 8` will set the acquisition mode to envelope and set the number of acquisitions for averaging to 8.

4. When you concatenate queries, the responses to all the queries are concatenated into a single response message. For example, if the display graticule is set to Full and the display style is set to dotsonly, the concatenated query `DISplay:GRAticule?;STYle?` will return the following.

   If the header is on:

   `DISPLAY:GRATICULE FULL;:DISPLAY:STYLE DOTSONLY`
If the header is off:

**FULL;DOTSONLY**

1. Set commands and queries may be concatenated in the same message. For example,

   
   `ACQuire:MODe SAMple;NUMAVg?;STATE?`

   is a valid message that sets the acquisition mode to sample. The message then queries the number of acquisitions for averaging and the acquisition state. Concatenated commands and queries are executed in the order received.

   Here are some invalid concatenations:

   `DISPlay:STYle DOTsonly OFF;ACQuire:NUMAVg 8` (no colon before `ACQuire`)

   `DISPlay:GRAticule FULL;:STYle DOTSONLY OFF` (extra colon before `STYle`)

   `DISPlay:GRAticule FULL;:*TRG` (colon before a star (*) command)

   `ACQUIRE:FASTACQ:PALETTE TEMPerature;FASTAcq:STATE ON` (levels of the mnemonics are different, either remove the second use of `FASTACQ:` or place `:ACQUIRE` in front of `FASTAcq:STATE`)

**Terminating**

This documentation uses \(<\text{EOM}\rangle\) (End of Message) to represent a message terminator.

**Table 2-4: End of Message Terminator**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;\text{EOM}\rangle)</td>
<td>Message terminator</td>
</tr>
</tbody>
</table>

The end-of-message terminator must be the END message (EOI asserted concurrently with the last data byte). The last data byte may be an ASCII line feed (LF) character.

This oscilloscope does not support ASCII LF only message termination. The oscilloscope always terminates outgoing messages with LF and EOI.

**Constructed Mnemonics**

Some header mnemonics specify one of a range of mnemonics. For example, a channel mnemonic can be CH1, CH2, CH3, CH4, CH5, CH6, CH7, or CH8 depending on the number of FlexChannels in your instrument. You use these mnemonics in the command just as you do any other mnemonic. For example, there is a CH1:POSITION command, and there is also a CH2:POSITION command. In the command descriptions, this list of choices is abbreviated as CH<i>. 
Command Syntax

**Bus Mnemonics**

Commands specify the bus to use as a mnemonic in the header.

Table 2-5: Bus Mnemonics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B&lt;x&gt;</td>
<td>A bus specifier; &lt;x&gt; is ≥1.</td>
</tr>
</tbody>
</table>

**Channel Mnemonics**

Commands specify the channel to use as a mnemonic in the header.

Table 2-6: Channel Mnemonics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH&lt;x&gt;</td>
<td>A channel specifier; &lt;x&gt; is 1 through 8 and is limited by the number of FlexChannels in your instrument.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_D&lt;x&gt;</td>
<td>A digital channel specifier; &lt;x&gt; in CH&lt;x&gt; is 1 through 8 and is limited by the number of FlexChannels in your instrument. &lt;x&gt; in D&lt;x&gt; is 0 through 7. Together they define a FlexChannel digital input.</td>
</tr>
</tbody>
</table>

**Cursor Position Mnemonics**

When cursors are displayed, commands may specify which cursor of the pair to use.

Table 2-7: Cursor Mnemonics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURSOR&lt;x&gt;</td>
<td>A cursor selector; &lt;x&gt; is must be 1 or 2.</td>
</tr>
</tbody>
</table>

**MathSpecifier Mnemonics**

Commands can specify the mathematical waveform to use as a mnemonic in the header.

Table 2-8: MathSpecifier Mnemonics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH&lt;x&gt;</td>
<td>A math waveform specifier; &lt;x&gt; is ≥1.</td>
</tr>
</tbody>
</table>

**Measurement Specifier Mnemonics**

Commands can specify which measurement to set or query as a mnemonic in the header.

Table 2-9: Measurement Specifier Mnemonics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEAS&lt;x&gt;</td>
<td>A measurement specifier; &lt;x&gt; is ≥1.</td>
</tr>
</tbody>
</table>
Commands can specify the reference waveform to use as a mnemonic in the header.

Table 2-10: Reference Waveform Mnemonics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>REF&lt;x&gt;</td>
<td>A reference waveform specifier; &lt;x&gt; is ≥1.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_D&lt;x&gt;</td>
<td>A digital reference waveform specifier; &lt;x&gt; is ≥1.</td>
</tr>
</tbody>
</table>

Commands can specify the view to use as a mnemonic in the header.

Table 2-11: Waveview Mnemonics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAVEView&lt;x&gt;</td>
<td>A waveview specifier; &lt;x&gt; must be equal to 1.</td>
</tr>
<tr>
<td>PLOTView&lt;x&gt;</td>
<td>A plotview specifier; &lt;x&gt; must be equal to 1.</td>
</tr>
<tr>
<td>MATHFFTView&lt;x&gt;</td>
<td>A mathfftview specifier; &lt;x&gt; must be equal to 1.</td>
</tr>
</tbody>
</table>

Commands can specify a search to use as a mnemonic in the header.

Table 2-12: Search Mnemonics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH&lt;x&gt;</td>
<td>A Search specifier; &lt;x&gt; is ≥1.</td>
</tr>
</tbody>
</table>

Commands can specify a zoom to use as a mnemonic in the header.

Table 2-13: Zoom Mnemonics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZOOM&lt;x&gt;</td>
<td>A zoom specifier; &lt;x&gt; must be equal to 1.</td>
</tr>
</tbody>
</table>

Commands use arguments such as enumeration, numeric, quoted string and block. Each of these arguments are listed in detail below.

**Enumeration**

Enter these arguments as unquoted text words. Like key words, enumeration arguments follow the same convention where the portion indicated in uppercase is required and that in lowercase is optional.

For example: ACQuire:MODe SAMPLE
Many oscilloscope commands require numeric arguments. The syntax shows the format that the oscilloscope returns in response to a query. This is also the preferred format when sending the command to the oscilloscope, though any of the formats will be accepted. This documentation represents these arguments as described below.

### Table 2-14: Numeric Arguments

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NR1&gt;</td>
<td>Signed integer value</td>
</tr>
<tr>
<td>&lt;NR2&gt;</td>
<td>Floating point value without an exponent</td>
</tr>
<tr>
<td>&lt;NR3&gt;</td>
<td>Floating point value with an exponent</td>
</tr>
<tr>
<td>&lt;bin&gt;</td>
<td>Signed or unsigned integer in binary format</td>
</tr>
</tbody>
</table>

Most numeric arguments will be automatically forced to a valid setting, by either rounding or truncating, when an invalid number is input, unless otherwise noted in the command description.

### Quoted String

Some commands accept or return data in the form of a quoted string, which is simply a group of ASCII characters enclosed by a single quote (') or double quote ("). The following is an example of a quoted string: "This is a quoted string". This documentation represents these arguments as follows:

### Table 2-15: Quoted String Argument

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;QString&gt;</td>
<td>Quoted string of ASCII text</td>
</tr>
</tbody>
</table>

A quoted string can include any character defined in the 7-bit ASCII character set. Follow these rules when you use quoted strings:

1. Use the same type of quote character to open and close the string. For example: "this is a valid string".
2. You can mix quotation marks within a string as long as you follow the previous rule. For example: "this is an 'acceptable' string".
3. You can include a quote character within a string by repeating the quote. For example: "here is a "" mark".  
4. Strings can have upper or lower case characters.
5. A carriage return or line feed embedded in a quoted string does not terminate the string. The return is treated as another character in the string.
6. The maximum length of a quoted string returned from a query is 1000 characters.
Here are some invalid strings:

- "Invalid string argument" (quotes are not of the same type)
- "test<EOI>" (termination character is embedded in the string)

### Block

Some commands use a block argument form to define a range or type of value, as defined in the table below.

#### Table 2-16: Block Argument

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NZDig&gt;</td>
<td>A nonzero digit character in the range of 1–9</td>
</tr>
<tr>
<td>&lt;Dig&gt;</td>
<td>A digit character, in the range of 0–9</td>
</tr>
<tr>
<td>&lt;DChar&gt;</td>
<td>A character with the hexadecimal equivalent of 00 through FF (0 through 255 decimal)</td>
</tr>
<tr>
<td>&lt;Block&gt;</td>
<td>A block of data bytes defined as: `&lt;Block&gt; ::= {#&lt;NZDig&gt;&lt;Dig&gt;[&lt;Dig&gt;...]</td>
</tr>
</tbody>
</table>

- `<NZDig>` specifies the number of `<Dig>` elements that follow. Taken together, the `<NZDig>` and `<Dig>` elements form a decimal integer that specifies how many `<DChar>` elements follow.
Command groups

The programmable interface conforms to Tektronix standard codes and formats except where noted. The interface also conforms to IEEE Std 488.2-1987 except where noted.

Acquisition command group

Acquisition commands set up the modes and functions that control how the instrument acquires signals and processes them into waveforms. Using these commands for acquiring waveforms, you can do the following:

- Start and stop acquisitions.
- Control whether each waveform is simply acquired, averaged, or enveloped over successive acquisitions of that waveform.
- Set the controls or conditions that start and stop acquisitions.
- Control acquisition of acquired channel waveforms.
- Set acquisition parameters.

Table 2-17: Acquisition commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQire?</td>
<td>Queries the current acquisition state.</td>
</tr>
<tr>
<td>ACQire:FASTAcq:PALEtte</td>
<td>Sets or queries the waveform grading for fast acquisition mode.</td>
</tr>
<tr>
<td>ACQire:FASTAcq:STATE</td>
<td>Sets or queries the state of fast acquisition mode.</td>
</tr>
<tr>
<td>ACQire:MAXSamplerate?</td>
<td>Returns the maximum real-time sample rate.</td>
</tr>
<tr>
<td>ACQire:MODe</td>
<td>Sets or queries the acquisition mode.</td>
</tr>
<tr>
<td>ACQire:NUMAcq?</td>
<td>Returns the number of waveform acquisitions that have occurred since starting acquisition with the ACQire:STATE_RUN command.</td>
</tr>
<tr>
<td>ACQire:NUMAVg</td>
<td>Sets or queries number of acquisitions for an averaged waveform.</td>
</tr>
<tr>
<td>ACQire:SEQUence:CURrent?</td>
<td>In single sequence acquisition mode, this query returns the number of acquisitions or measurements in the sequence completed so far.</td>
</tr>
<tr>
<td>ACQire:SEQUence:MODEe</td>
<td>In single sequence acquisition, the single sequence stop after count is based on number of acquisitions or measurements. Number of acquisitions is the only mode supported for this product.</td>
</tr>
<tr>
<td>ACQire:SEQUence:NUMSEQUence</td>
<td>In single sequence acquisition mode, specify the number of acquisitions or measurements that comprise the sequence.</td>
</tr>
<tr>
<td>ACQire:STATE</td>
<td>Starts, stops, or returns acquisition state.</td>
</tr>
<tr>
<td>ACQire:STOPAfter</td>
<td>Sets or queries whether the acquisition is continuous or single sequence.</td>
</tr>
</tbody>
</table>
AFG Command Group

Use the AFG commands for Arbitrary Function Generator functionality. Requires option AFG.

Table 2-18: AFG commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFG:AMPLitude</td>
<td>Sets (or queries) the AFG amplitude in volts, peak to peak.</td>
</tr>
<tr>
<td>AFG:ARBitrary:SOUrce</td>
<td>Sets or queries the source name for the Arbitrary Waveform.</td>
</tr>
<tr>
<td>AFG:FREQuency</td>
<td>Sets (or queries) the AFG frequency, in Hz.</td>
</tr>
<tr>
<td>AFG:FUNCtion</td>
<td>Sets (or queries) which AFG function to execute.</td>
</tr>
<tr>
<td>AFG:HIGHLevel</td>
<td>Sets (or queries) the high level value of the output waveform, in volts.</td>
</tr>
<tr>
<td>AFG:LOWLevel</td>
<td>Sets (or queries) the low level value of the output waveform, in volts.</td>
</tr>
<tr>
<td>AFG:NOISEAdd:PERCent</td>
<td>Sets (or queries) the AFG additive noise level as a percentage.</td>
</tr>
<tr>
<td>AFG:NOISEAdd:STATE</td>
<td>Sets (or queries) the AFG additive noise state.</td>
</tr>
<tr>
<td>AFG:OFFSet</td>
<td>Sets (or queries) the AFG offset value, in volts.</td>
</tr>
<tr>
<td>AFG:OUTPut:LOAd:IMPEDance</td>
<td>Sets (or queries) the AFG output load impedance.</td>
</tr>
<tr>
<td>AFG:OUTPut:STATE</td>
<td>Sets (or queries) the AFG output state.</td>
</tr>
<tr>
<td>AFG:PERIod</td>
<td>Sets (or queries) the period of the AFG waveform, in seconds.</td>
</tr>
<tr>
<td>AFG:PULse:WIDth</td>
<td>Sets (or queries) the AFG pulse width, in seconds.</td>
</tr>
<tr>
<td>AFG:RAMP:SYMmetry</td>
<td>Sets (or queries) the AFG ramp symmetry as a percentage.</td>
</tr>
<tr>
<td>AFG:SQUare:DUty</td>
<td>Sets (or queries) the AFG duty cycle, as a percentage.</td>
</tr>
</tbody>
</table>
Alias command group

Alias commands allow you to define new commands as a sequence of standard commands. You might find this useful when repeatedly using the same commands to perform certain tasks like setting up measurements.

Aliases are similar to macros but do not include the capability to substitute parameters into alias bodies. The alias mechanism obeys the following rules:

- The alias name must consist of a valid IEEE 488.2 message unit, which may not appear in a message preceded by a colon, comma, or a command or query program header.
- The alias name may not appear in a message followed by program date, a colon, comma, or question mark.
- An alias name must be distinct from any keyword or keyword short form.
- An alias name cannot be redefined without first being deleted using one of the alias deletion functions.
- Alias names do not appear in response messages.
- The Alias commands are defined in Tektronix Standard Codes and Formats. Deviations between that standard and what is specified here will be considered errors unless specifically noted in the command description in this document.

Table 2-19: Alias commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALias</td>
<td>Sets or queries the alias state.</td>
</tr>
<tr>
<td>ALias:CATalog?</td>
<td>Returns a list of the currently defined alias labels.</td>
</tr>
<tr>
<td>ALias:DEFine</td>
<td>Assigns a sequence of program messages to an alias label.</td>
</tr>
<tr>
<td>ALias:DELEte</td>
<td>Removes a specified alias.</td>
</tr>
<tr>
<td>ALias:DELEte:ALL</td>
<td>Deletes all existing aliases.</td>
</tr>
<tr>
<td>ALias:DELEte:NAME</td>
<td>Removes a specified alias.</td>
</tr>
<tr>
<td>ALias:STATE</td>
<td>Sets or queries the alias state.</td>
</tr>
</tbody>
</table>
Bus command group

Use the commands in the Bus Command Group to configure a bus. These commands let you:

- Specify the bus type.
- Specify the signals to be used in the bus.
- Specify its display style.

**NOTE.** Bus commands are present once a bus has been added.

Bus Mnemonics

Commands specify the bus to use as a mnemonic in the header.

Table 2-20: Bus mnemonics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B&lt;x&gt;</td>
<td>A bus specifier; &lt;x&gt; is ≥1.</td>
</tr>
</tbody>
</table>

Table 2-21: Bus commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS:ADDNew</td>
<td>Adds the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ARINC429A:SOURc</td>
<td>Sets or queries the source for the specified ARINC429 bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ARINC429A:THRESHold</td>
<td>Sets or queries the ARINC429 upper threshold for the specified ARINC429 bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ARINC429A:POLARITY</td>
<td>Sets or queries the source polarity for the specified ARINC429 bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ARINC429A:DATAFORmat</td>
<td>Sets or queries the format of the DATA field for the specified ARINC429 bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ARINC429A:BITRate</td>
<td>Sets or queries the ARINC429 bit rate for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ARINC429A:BITRate:CUSTom</td>
<td>Sets or queries the ARINC429 custom bit rate for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:BITDelay</td>
<td>Sets or queries the number of delay bits for the specified AUDIO bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:BITOrder</td>
<td>Specifies the bit order for the specified AUDIO bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:CLOCk:POLarity</td>
<td>Sets or queries the clock polarity for the specified AUDIO bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:CLOCk:SOURc</td>
<td>Sets or queries the clock source waveform for the specified AUDIO bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:CLOCk:THReshold</td>
<td>Sets or queries the Audio Clock source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:DATA:POLarity</td>
<td>Specifies the data polarity for the specified AUDIO bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:DATA:SIZe</td>
<td>Specifies the number of bits per word for the specified AUDIO bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:DATA:SOURc</td>
<td>Specifies the audio data source waveform for the specified AUDIO bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:DATA:THReshold</td>
<td>Sets or queries the Audio Data source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:DATA:WORDSIZe</td>
<td>Sets or queries the Audio bits per word for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:FRAME: CLOCKBITSPERCHANNEL</td>
<td>Sets or queries the Audio bits of sync width for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:FRAME:SIZe</td>
<td>Specifies the number of channels in each frame for the specified AUDIO bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:TYPe</td>
<td>Specifies the audio format (type) for the specified AUDIO bus.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:WORDSel:POLarity</td>
<td>Sets or queries the word select polarity for the specified AUDIO bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:WORDSel:SOUrce</td>
<td>Specifies the word select source waveform for the AUDIO bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:WORDSel:THReshold</td>
<td>Sets or queries the Audio Word Select source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:CAN:BITRate</td>
<td>Sets or queries the CAN bit rate.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:CAN:BITRate:VALue</td>
<td>Sets or queries CAN bit rate.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:CAN:FD:BITRate</td>
<td>Sets or queries the increased data phase bit rate used by CAN FD packets on the specified CAN bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:CAN:FD:BITRate:CUSTom</td>
<td>Sets or queries the custom bit rate for the increased data phase of CAN FD packets on the specified CAN bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:CAN:SAMPLEpoint</td>
<td>Sets or queries the sample point for the specified CAN bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:CAN:SIGNAL</td>
<td>Sets or queries the signal type for the specified CAN bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:CAN:SOUrce</td>
<td>Sets or queries the CAN source channel.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:CAN:STANDard</td>
<td>Sets or queries which CAN standard specification to analyze the specified CAN bus with.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:CAN:THReshold</td>
<td>Sets or queries the source channel threshold for the specified CAN bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:DISplay:FORMat</td>
<td>Sets or queries how the data is represented in the busform for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:DISplay:LAYout</td>
<td>This command sets or queries the format a bus layer should use.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ETHERnet:DATAMINUSTHRESHold</td>
<td>Sets or queries the Ethernet DATA Minus source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ETHERnet:DATAPLUSTHRESHold</td>
<td>Sets or queries the Ethernet DATA Plus source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ETHERnet:IPVFOUR</td>
<td>Sets or queries whether IPV4 packets are available for triggering on Ethernet.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ETHERnet:LOWTHRESHold</td>
<td>Sets or queries the Ethernet DATA source Low threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ETHERnet:QTAGGING</td>
<td>Sets or queries whether Q-Tagging packets are available for triggering on Ethernet.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ETHERnet:SIGNALTYpe</td>
<td>Sets or queries the Ethernet signal type for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ETHERnet:SOUrce</td>
<td>Specifies the Ethernet data source for differential input.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ETHERNET:SOUrce:DMINus</td>
<td>Sets or queries the Ethernet DMINus source.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ETHERNET:SOUrce:DPLUS</td>
<td>Sets or queries the Ethernet DPLUS source.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ETHERnet:THReshold</td>
<td>Sets or queries the Ethernet DATA source High threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:ETHERnet:TYPE</td>
<td>Specifies the Ethernet standard type: 10Base-T or 100Base-T.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:FLEXray:BITRate</td>
<td>Sets or queries the FlexRay bus bit rate.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:FLEXray:BITRate:CUSTom</td>
<td>Sets or queries the FlexRay custom bit rate for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:FLEXray:Channel</td>
<td>Sets or queries the FlexRay bus channel.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:FLEXray:LOWTHRESHold</td>
<td>Sets or queries the FlexRay data source low threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:FLEXray:SIGNAL</td>
<td>Sets or queries the FlexRay probe.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:FLEXray:SOUrce</td>
<td>Sets or queries the FlexRay bus source.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:FLEXray:SOUrce:TXRX</td>
<td>Sets or queries the FlexRay TxRx data source for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:FLEXray:THReshold</td>
<td>Sets or queries the FlexRay data source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:FLEXray:TXRXTHRESHold</td>
<td>Sets or queries the FlexRay data source TxRx threshold for the specified bus.</td>
</tr>
</tbody>
</table>
### Table 2-21: Bus commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BUS:B&lt;x&gt;:I2C:CLOCk:SOUrce</code></td>
<td>Sets or queries the I2C clock (SCLK) source for the specified bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:I2C:CLOCk:THReshold</code></td>
<td>Sets or queries the I2C Clock (SCLK) source threshold for the specified bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:I2C:DATa:SOUrce</code></td>
<td>Sets or queries the I2C data (SDA) source for the specified bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:I2C:DATa:THReshold</code></td>
<td>Sets or queries the I2C Data (SDA) source threshold for the specified bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:I2C:RWINADDR</code></td>
<td>Determines whether decoded I2C slave addresses are pure seven-bit values, or have the R/W* combined with them.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LABel:COLor</code></td>
<td>Sets or queries the color of the specified bus label.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LABel:FONT:BOLD</code></td>
<td>Sets or queries the bold state of the specified bus label.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LABel:FONT:ITALic</code></td>
<td>Sets or queries the italic state of the specified bus label.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LABel:FONT:SIZE</code></td>
<td>Sets or queries the font size of the specified bus label.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LABel:FONT:TYPE</code></td>
<td>Sets or queries the font type of the specified bus label, such as Arial or Times New Roman.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LABel:FONT:UNDERline</code></td>
<td>Sets or queries the underline state of the specified bus label.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LABel:name</code></td>
<td>Sets or queries the waveform label for the specified bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LABel:XPOS</code></td>
<td>Sets or queries the x-position of the specified bus' label.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LABel:YPOS</code></td>
<td>Sets or queries the y-position of the specified bus' label.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LIN:BITRate</code></td>
<td>Sets or queries the LIN bus bit rate.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LIN:IDFORmat</code></td>
<td>Sets or queries LIN bus id format.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LIN:POLarity</code></td>
<td>Sets or queries the LIN bus polarity.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LIN:SAMPLEpoint</code></td>
<td>Specifies the point to sample during each bit period, as a percent, for the specified LIN bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LIN:SOUrce</code></td>
<td>Sets or queries sets the LIN bus source.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LIN:SOUrce:THReshold</code></td>
<td>Sets or queries the LIN source threshold for the specified bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LIN:STANDard</code></td>
<td>Sets or queries the LIN bus standard.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:LIN:BITRate:CUSTom</code></td>
<td>Sets or queries LIN custom bit rate for the specified bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:MIL1553B:SOUrce</code></td>
<td>Sets or queries the source for the specified MIL-STD-1553 bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:MIL1553B:THRESHold</code></td>
<td>Sets or queries the MIL-STD-1553 upper threshold for the specified bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:MIL1553B:LOWTHRESHHold</code></td>
<td>Sets or queries the MIL-STD-1553 lower threshold for the specified bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:MIL1553B:POLarity</code></td>
<td>Sets or queries the source polarity for the specified MIL-STD-1553 bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:MIL1553B:RESPonsetime:MINimum</code></td>
<td>Sets or queries the minimum response time to a valid command issued for the specified MIL-STD-1553 bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:MIL1553B:RESPonsetime:MAXimum</code></td>
<td>Sets or queries the maximum response time to a valid command issued for the specified MIL-STD-1553 bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:PARallel:ALLTHResholds</code></td>
<td>Sets or queries the threshold for all sources for the parallel bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:PARallel:ALLTHResholds:APPlY</code></td>
<td>Sets all of the data source thresholds to the value of the allMRefs parameter for the parallel bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:PARallel:BIT&lt;n&gt;SOUrce</code></td>
<td>Sets or queries the specified bit source for specified parallel bus.</td>
</tr>
<tr>
<td><code>BUS:B&lt;x&gt;:PARallel:BIT&lt;n&gt;THReshold</code></td>
<td>Sets or queries the specified bit source threshold for the specified parallel bus.</td>
</tr>
</tbody>
</table>
Table 2-21: Bus commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS:B&lt;x&gt;:PARallel:CLOCk:EDGE</td>
<td>Determines which edges of its clock signal cause a clocked parallel bus to sample new states.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:PARallel:CLOCk:ISCLOCKED</td>
<td>Determines whether the bus operates in a clocked or asynchronous fashion.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:PARallel:CLOCkSOUrce</td>
<td>Sets or queries the Parallel clock source for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:PARallel:CLOCkSOUrce:THReshold</td>
<td>Sets or queries the bit source threshold for the parallel bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:BITRate</td>
<td>Sets or queries the RS-232 bit rate for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:BITRate:CUSTom</td>
<td>Sets or queries the RS232 custom bit rate for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:DATABits</td>
<td>Sets or queries the number of RS-232 data bits for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:DELIMiter</td>
<td>Sets or queries the RS-232 delimiting value for a packet on the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:DISplaymode</td>
<td>Sets or queries the display mode for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:PARity</td>
<td>Sets or queries the RS-232 parity for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:SOUrce</td>
<td>Sets or queries the RS-232 polarity for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:SOUrce:THReshold</td>
<td>Sets or queries the RS232 source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SENT:CHANWidth</td>
<td>Sets or queries SENT fast channel bit widths for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SENT:NIBBLECount</td>
<td>Sets or queries SENT data nibbles for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SENT:NUMCHANnel</td>
<td>Sets or queries SENT fast data channels for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SENT:PAUSEPULSe</td>
<td>Sets or queries SENT pause pulse for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SENT:POLARITY</td>
<td>Sets or queries SENT Idle State signal polarity for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SENT:SLOW</td>
<td>Sets or queries the SENT slow channel configuration for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SENT:SOUrce</td>
<td>Sets or queries the SENT DATA source for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SENT:THRESHold</td>
<td>Sets or queries the SENT DATA source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SENT:TICKTIME</td>
<td>Sets or queries the SENT bus Clock Tick parameter for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SENT:TICKTOLerance</td>
<td>Sets or queries the SENT bus Tick Tolerance percent parameter for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:BITOrder</td>
<td>Sets or queries the shift direction used to de-serialize data for the SPI mode of the bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:CLOCk:POLarity</td>
<td>Sets or queries the SPI clock (SCLK) polarity for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:CLOCk:SOUrce</td>
<td>Sets or queries the SPI clock (SCLK) source for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:CLOCk:THReshold</td>
<td>Sets or queries the SPI Clock (SCLK) source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:DATa:POLarity</td>
<td>Sets or queries the SPI data (DATA) polarity for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:DATa:SIZe</td>
<td>Sets or queries the number of bits per word for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:DATa:SOUrce</td>
<td>Sets or queries the SPI data (DATA) source for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:DATa:THReshold</td>
<td>Sets or queries the SPI DATA source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:FRAMING</td>
<td>Sets or queries the SPI bus framing.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:IDLETime</td>
<td>Sets or queries the SPI bus idle time.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:SELect:POLarity</td>
<td>Sets or queries the SPI Slave Select (SS) polarity for the specified bus.</td>
</tr>
</tbody>
</table>
**Table 2-21: Bus commands (cont.)**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS:B&lt;x&gt;:SPI:SELect:SOUrce</td>
<td>Sets or queries the SPI Slave Select (SS) source for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:SELect:THReshold</td>
<td>Sets or queries the SPI Select (SS) source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPMI:SCLk:SOUrce</td>
<td>This command sets or queries the SPMI Clock (SCLK) source for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPMI:SCLk:THReshold</td>
<td>This command sets or queries the SPMI Clock (SCLK) source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPMI:SDATa:SOUrce</td>
<td>This command sets or queries the SPMI Data (SDATA) source for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPMI:SDATa:THReshold</td>
<td>This command sets or queries the SPMI Data (SDATA) source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:TYPe</td>
<td>Sets or queries the bus type specified.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:USB:BITRate</td>
<td>Sets or queries the USB bit rate for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:USB:SOUrce</td>
<td>Sets or queries the USB Data Source for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:USB:SOUrce:DMINus</td>
<td>Sets or queries the USB Data Source for D- input for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:USB:SOUrce:DPLUS</td>
<td>Sets or queries the USB Data Source for D+ input for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:USB:DATAMINUSTHRESHold</td>
<td>Sets or queries the USB DATA Minus source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:USB:DATAPLUSTHRESHold</td>
<td>Sets or queries the USB DATA Plus source threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:USB:LOWTHRESHold</td>
<td>Sets or queries the USB DATA source Low threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:USB:SIGNALTyPe</td>
<td>Sets or queries the USB signal type for the specified bus.</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:USB:THRESHold</td>
<td>Sets or queries the USB DATA source High threshold for the specified bus.</td>
</tr>
<tr>
<td>BUS:DELete</td>
<td>Deletes the specified bus.</td>
</tr>
<tr>
<td>BUS:LIST?</td>
<td>Lists all currently defined bus.</td>
</tr>
<tr>
<td>BUSTABle:ADDNew</td>
<td>Adds the specified bus table.</td>
</tr>
<tr>
<td>BUSTABle:DELete</td>
<td>Deletes the specified bus table.</td>
</tr>
<tr>
<td>BUSTABle:LIST?</td>
<td>Lists all currently defined bus tables.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:BUS:B&lt;x&gt;:STATE</td>
<td>Sets or queries the state of the specified bus in the specified Waveview.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:BUS:B&lt;x&gt;:VERTical:POSition</td>
<td>Sets or queries the vertical position of the specified bus in the specified Waveview.</td>
</tr>
</tbody>
</table>
Calibration command group

The Calibration commands provide information about the current state of instrument calibration and allow you to initiate signal path calibration (SPC).

**NOTE.** When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes, which can take several minutes. Any remote command that performs an action on the oscilloscope is also disabled until the SPC is complete.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*CAL?</td>
<td>Instructs the instrument to perform signal path calibration and returns the calibration status when complete. Takes several minutes to run.</td>
</tr>
<tr>
<td>CALibrate?</td>
<td>Returns the calibration status.</td>
</tr>
<tr>
<td>CALibrate:INTERNal</td>
<td>Starts the signal path calibration. Takes several minutes to run.</td>
</tr>
<tr>
<td>CALibrate:INTERNal:START</td>
<td>Starts the signal path calibration.</td>
</tr>
<tr>
<td>CALibrate:INTERNal:STATus?</td>
<td>Returns the status of the signal path calibration.</td>
</tr>
<tr>
<td>CALibrate:PWRUpstatus?</td>
<td>Returns the current status of the power-up calibration.</td>
</tr>
<tr>
<td>TOUCHSCReen:CALibrate</td>
<td>Launches the touchscreen calibration.</td>
</tr>
<tr>
<td>TOUCHSCReen:STATe</td>
<td>Sets or queries the enabled state of the touch screen.</td>
</tr>
</tbody>
</table>
Cursor command group

Use the commands in the Cursor Command Group to control the cursor display and readout. You can use these commands to control the setups for each cursor, such as waveform source, and cursor position.

You can also use the commands to select one of the following cursor functions:

- **Off.** Shuts off the display of all cursors.
- **Vertical bars.** Displays vertical bar cursors, which provide traditional horizontal unit readouts for Cursor 1 (bar1), Cursor 2 (bar2), the delta between them, and 1/delta (results in frequency when the horizontal unit is time). Vertical bars are another name for vertical screen cursors.
- **Horizontal bars.** Displays horizontal bar cursors, which provide traditional vertical unit readouts for Cursor 1 (bar1), Cursor 2 (bar2), and the delta between them. Horizontal bars are another name for horizontal screen cursors.
- **Waveform cursors.** Consists of two cursors you can independently assign to a waveform. Waveform cursors enable you to conveniently measure waveform amplitude and time at specific points on the waveform. In XY or XYZ format, waveform cursors indicate the amplitude position of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis and Ch2 is the Y axis) relative to the trigger.
- **Screen cursors.** Consist of two pairs of horizontal and vertical bar cursors. You can use these cursors to indicate an arbitrary position within the waveform display area. Screen cursors are basically just turning on horizontal bars and vertical bars at the same time. These cursors have no association with any waveform, except that they inherit the color of the waveform they are assigned to.

**NOTE.** Cursor commands are available once a view has been added.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display:MATHFFTView&lt;x&gt;:CURSor:A SOURCE?</td>
<td>Queries the cursor source for cursor A in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>Display:MATHFFTView&lt;x&gt;:CURSor:B SOURCE?</td>
<td>Queries the cursor source for cursor B in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>Display:MATHFFTView&lt;x&gt;:CURSor:DDT?</td>
<td>Queries the delta V over delta T cursor readout value in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>Display:MATHFFTView&lt;x&gt;:CURSor:FUNCtion</td>
<td>Sets or queries the cursor typein the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>Display:MATHFFTView&lt;x&gt;:CURSor:HBARs:APOSITION</td>
<td>Sets or returns the horizontal cursor A position in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:HBArs:AUNits?</td>
<td>Queries cursor A vertical units of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:HBArs:BPOSition</td>
<td>Sets or returns the vertical cursor B position in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:HBArs:BUNIts?</td>
<td>Queries the cursor B vertical units in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:HBArs:DELTa?</td>
<td>Queries the delta V cursor readout value of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:MODe</td>
<td>Sets or returns the cursor tracking mode in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:ONEOVERDELTATVALUE?</td>
<td>Queries the one over delta T cursor readout value in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:SCREEN:AXPOSition</td>
<td>Sets or returns the horizontal cursor A position in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:SCREEN:AYPOSition</td>
<td>Sets or returns the vertical cursor A position in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:SCREEN:BXPOSition</td>
<td>Sets or returns the horizontal cursor B position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:SCREEN:BYPOSition</td>
<td>Sets or returns the vertical cursor B position in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:STATE</td>
<td>Sets or queries the visible state of cursors in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:VBArs:APOSition</td>
<td>Queries the vertical cursor A position in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:VBArs:BPOSition</td>
<td>Sets or queries the horizontal cursor B position in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:VBArs:BUNIts?</td>
<td>Queries the vertical cursor B measurement units for the specified Math-FFT view.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:VBArs:DELTa?</td>
<td>Queries the delta T cursor readout value of the specified cursor in the specified Math-FFT waveform.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:WAVEform:APOSition</td>
<td>Sets or queries the waveform cursor A position in the specified plot view.</td>
</tr>
<tr>
<td>DISplay:MATHFFTView&lt;x&gt;:CURSor:WAVEform:BPOSition</td>
<td>Sets or queries the waveform cursor B position in the specified plot view.</td>
</tr>
<tr>
<td>DISplay:PLOTView&lt;x&gt;:CURSor:ASOURce?</td>
<td>Queries the cursor source for plot cursor A.</td>
</tr>
<tr>
<td>DISplay:PLOTView&lt;x&gt;:CURSor:BSOURce?</td>
<td>Queries the cursor source for plot cursor B.</td>
</tr>
<tr>
<td>DISplay:PLOTView&lt;x&gt;:CURSor:DDT?</td>
<td>Queries the delta V over delta T cursor readout value of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>DISplay:PLOTView&lt;x&gt;:CURSor:FUNCTION</td>
<td>Sets or queries the cursor mode of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>DISplay:PLOTView&lt;x&gt;:CURSor:HBArs:APOSition</td>
<td>Sets or queries the vertical cursor A position of the specified cursor in the specified view.</td>
</tr>
</tbody>
</table>
Table 2-23: Cursor commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:HBAr:s: AUNIts?</td>
<td>Sets or queries the cursor A vertical units of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:HBAr:s: BPOSition</td>
<td>Sets or queries the vertical cursor B position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:HBAr:s: BUNIts?</td>
<td>Sets or queries the cursor B vertical units of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:HBAr:s: DELTa?</td>
<td>Sets or queries the delta V cursor readout value of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:MODe</td>
<td>Sets or queries the cursor tracking mode of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:ONEOVERDELTATVALUE?</td>
<td>Sets or queries the one over delta T cursor readout value of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:SCREEN::APOSition</td>
<td>Sets or queries the horizontal cursor A position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:SCREEN::AYPOSition</td>
<td>Sets or queries the vertical cursor A position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:SCREEN::BXPOSition</td>
<td>Sets or queries the horizontal cursor B position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:SCREEN::BYPOSition</td>
<td>Sets or queries the vertical cursor B position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:SPLITMODE</td>
<td>Sets or queries the cursor source mode in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:STATE</td>
<td>Sets or queries the visible state of the cursor of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:VBAr:s: APOSition</td>
<td>Sets or queries the horizontal cursor A position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:VBAr:s: BPOSition</td>
<td>Sets or queries the horizontal cursor B position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:VBAr:s: DELTa?</td>
<td>Queries the delta T cursor readout value of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:VBAr:s: UNIts?</td>
<td>Queries the VBArs cursor readout units of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:WAVeform::APOSition</td>
<td>Sets or queries the horizontal cursor A position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: PLOTView&lt;x&gt;::CURSor:WAVeform::BPOSition</td>
<td>Sets or queries the horizontal cursor B position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: REFFFTView&lt;x&gt;::CURSor::ASource?</td>
<td>Queries the cursor source for plot cursor A</td>
</tr>
<tr>
<td>Display: REFFFTView&lt;x&gt;::CURSor::BSOurc?</td>
<td>Queries the cursor source for plot cursor B.</td>
</tr>
<tr>
<td>Display: REFFFTView&lt;x&gt;::CURSor:DDT?</td>
<td>Queries the delta V over delta T cursor readout value of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Display: REFFFTView&lt;x&gt;::CURSor:FUNCtion</td>
<td>Sets or queries the cursor type of the specified cursor in the specified view.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:HBArs:APOSition</strong></td>
<td>Sets or queries the vertical cursor A position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:HBArs:AUNIts?</strong></td>
<td>Queries cursor A vertical units of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:HBArs:BPOSition</strong></td>
<td>Sets or queries the vertical cursor B position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:HBArs:BUUnIts?</strong></td>
<td>Queries the cursor B vertical units of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:HBArs:DELTa?</strong></td>
<td>Queries the delta V cursor readout value of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:HBArs:MODE</strong></td>
<td>Sets or queries the cursor tracking mode of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:HBArs:ONEOVERDELTATVALUE?</strong></td>
<td>Queries the one over delta T cursor readout value of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:SCREEN:AXPOSition</strong></td>
<td>Sets or queries the horizontal cursor A position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:SCREEN:AYPOSition</strong></td>
<td>Sets or queries the vertical cursor A position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:SCREEN:BXPOSition</strong></td>
<td>Sets or queries the horizontal cursor B position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:SCREEN:BYPOSition</strong></td>
<td>Sets or queries the vertical cursor B position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:SPLITMODE</strong></td>
<td>Sets or queries whether both cursors have same or different source.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:STATE</strong></td>
<td>Sets or queries the visible state of the cursor of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:VBArs:APOSition</strong></td>
<td>Sets or queries the horizontal cursor A position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:VBArs:BPOSition</strong></td>
<td>Sets or queries the horizontal cursor B position of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:VBArs:DELTa?</strong></td>
<td>Queries the delta T cursor readout value of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:VBArs:UNIts?</strong></td>
<td>Queries cursor A vertical units of the specified cursor in the specified view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:WAVEform:AHPOSition</strong></td>
<td>Queries the value of the cursor A horizontal position.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:WAVEform:APOSition</strong></td>
<td>Sets or returns the waveform cursor A position in the specified plot view.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:WAVEform:AVPOSition</strong></td>
<td>Queries the value of the cursor A vertical position.</td>
</tr>
<tr>
<td><strong>DISplay:REFFFTView&lt;x&gt;:CURSor:WAVEform:BHPOSition?</strong></td>
<td>Queries the value of the cursor B horizontal position.</td>
</tr>
</tbody>
</table>
### Table 2-23: Cursor commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{DISplay:REFFFTView&lt;x&gt;:CURSor:WAVEform:BPOSITION}</td>
<td>Sets or returns the waveform cursor B position in the specified plot view.</td>
</tr>
<tr>
<td>\texttt{DISplay:REFFFTView&lt;x&gt;:CURSor:WAVEform:BVPOSITION}</td>
<td>Queries the value of the cursor B vertical position.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor?}</td>
<td>Queries the cursor parameters for the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;;}</td>
<td>Queries the cursor parameters for the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:ASOURce}</td>
<td>Sets or queries the cursor A source of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:BSOURce}</td>
<td>Sets or queries the cursor B source of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:DDT?}</td>
<td>Returns the delta V over delta T cursor readout value of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:FUNCtion}</td>
<td>Sets or queries the cursor mode (SCREEN or DATA) of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:HBAr:sAPOSITION}</td>
<td>Sets or queries the vertical cursor A position of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:HBAr:sAUNits?}</td>
<td>Queries the cursor A vertical units of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:HBAr:sBPOSITION}</td>
<td>Sets or queries the vertical cursor B position of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:HBAr:sBUNits?}</td>
<td>Queries the cursor B vertical units of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:ONEOVERDELTATVALUE?}</td>
<td>Sets or queries the one over delta T cursor readout value of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:MODe}</td>
<td>Sets or queries the cursor tracking mode of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:ONEOVERDELTATVALUE?}</td>
<td>Sets or queries the one over delta T cursor readout value of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:SCREEn:AXPOSITION}</td>
<td>Sets or queries the horizontal cursor A position of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:SCREEn:AYPOSITION}</td>
<td>Sets or queries the vertical cursor A position of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:SCREEn:BXPOSITION}</td>
<td>Sets or queries the horizontal cursor B position of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:SCREEn:BYPOSITION}</td>
<td>Sets or queries the vertical cursor B position of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:SPLITMODe}</td>
<td>Sets or queries the cursor source mode in the specified view.</td>
</tr>
<tr>
<td>\texttt{DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:STATE}</td>
<td>Sets or queries the visible state of the cursor of the specified cursor in the specified waveview.</td>
</tr>
</tbody>
</table>
Table 2-23: Cursor commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:VBAr:s:APOSITION</td>
<td>Sets or queries the horizontal cursor A position of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:VBAr:s:BPOSITION</td>
<td>Sets or queries the horizontal cursor B position of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:VBAr:s:DELTa?</td>
<td>Sets or queries the delta T cursor readout value of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:VBAr:s:UNIts?</td>
<td>Queries cursor A vertical units of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:WAVEform:APOSITION</td>
<td>Sets or queries the horizontal cursor A position of the specified cursor in the specified waveview.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:CURSor:CURSOR&lt;x&gt;:WAVEform:BPOSITION</td>
<td>Sets or queries the horizontal cursor B position of the specified cursor in the specified waveview.</td>
</tr>
</tbody>
</table>
**Digital command group**

Use the commands in the Digital Command Group to acquire up to 64 digital signals and analyze them. Digital channels are only available when a digital probe is attached to the super channel.

Table 2-24: Digital commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH&lt;x&gt;_DALL:LABel:COLor</td>
<td>Sets or queries the color of the specified digital group label.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_DALL:LABel:FONT:BOLD</td>
<td>Sets or queries the bold state of the specified digital group.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_DALL:LABel:FONT:ITALic</td>
<td>Sets or queries the italic state of the specified digital group.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_DALL:LABel:FONT:SIZE</td>
<td>Sets or queries the font size of the specified digital group.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_DALL:LABel:FONT:TYPE</td>
<td>Sets or queries the font type of the specified digital group, such as Arial or Times New Roman.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_DALL:LABel:FONT:UNDERline</td>
<td>Sets or queries the underline state of the specified digital group.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_DALL:LABel:NAMe</td>
<td>Sets or queries the label of the specified digital group.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_D&lt;x&gt;:LABel:COLor</td>
<td>Sets or queries the color of the label of the specified digital bit.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_D&lt;x&gt;:LABel:FONT:BOLD</td>
<td>Sets or queries the bold state of the label of the specified digital bit.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_D&lt;x&gt;:LABel:FONT:ITALic</td>
<td>Sets or queries the italic state of the label of the specified digital bit.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_D&lt;x&gt;:LABel:FONT:SIZE</td>
<td>Sets or queries the font size of the label of the specified digital bit.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_D&lt;x&gt;:LABel:FONT:TYPE</td>
<td>Sets or queries the font type of the label of the specified digital bit.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_D&lt;x&gt;:LABel:FONT:UNDERline</td>
<td>Sets or queries the underline state of the label of the specified digital bit.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_D&lt;x&gt;:LABel:NAMe</td>
<td>Sets or queries the label of the specified digital bit.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_D&lt;x&gt;:LABel:XPOS</td>
<td>Sets or queries the x-position of the label of the specified digital bit.</td>
</tr>
<tr>
<td>CH&lt;x&gt;_D&lt;x&gt;:LABel:YPOS</td>
<td>Sets or queries the y-position of the label of the specified digital bit.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_DALL:LABel:COLor</td>
<td>Sets or queries the color of the specified digital group.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_DALL:LABel:FONT:BOLD</td>
<td>Sets or queries the bold state of the specified digital group.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_DALL:LABel:FONT:ITALic</td>
<td>Sets or queries the italic state of the specified digital group.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_DALL:LABel:FONT:SIZE</td>
<td>Sets or queries the font size of the specified digital group.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_DALL:LABel:FONT:TYPE</td>
<td>Sets or queries the font type of the specified digital group.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_DALL:LABel:FONT:UNDERline</td>
<td>Sets or queries the underline state of the specified digital group.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_DALL:LABel:NAMe</td>
<td>Sets or queries the label of the specified digital group.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_DALL:LABel:XPOS</td>
<td>Sets or queries the x-position of the label of the specified digital group.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_DALL:LABel:YPOS</td>
<td>Sets or queries the y-position of the label of the specified digital group.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_D&lt;x&gt;:LABel:COLor</td>
<td>Sets or queries the color of the label of the specified digital channel.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_D&lt;x&gt;:LABel:FONT:BOLD</td>
<td>Sets or queries the bold state of the label of the specified digital bit.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_D&lt;x&gt;:LABel:FONT:ITALic</td>
<td>Sets or queries the italic state of the label of the specified digital bit.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_D&lt;x&gt;:LABel:FONT:SIZE</td>
<td>Sets or queries the font size of the label of the specified digital bit.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_D&lt;x&gt;:LABel:FONT:TYPE</td>
<td>Sets or queries the font type of the label of the specified digital bit.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_D&lt;x&gt;:LABel:FONT:UNDERline</td>
<td>Sets or queries the underline state of the label of the specified digital bit.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_D&lt;x&gt;:LABel:NAMe</td>
<td>Sets or queries the label of the specified digital bit.</td>
</tr>
<tr>
<td>REF&lt;x&gt;_D&lt;x&gt;:LABel:XPOS</td>
<td>Sets or queries the x-position of the label of the specified digital bit.</td>
</tr>
</tbody>
</table>
Table 2-24: Digital commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REF&lt;x&gt;_D&lt;x&gt;:LABEL:YPOS</td>
<td>Sets or queries the y-position of the label of the specified digital channel.</td>
</tr>
<tr>
<td>DIGGRP&lt;x&gt;:D&lt;x&gt;:THreshold</td>
<td>Sets or queries the threshold level in volts for specified digital channel.</td>
</tr>
</tbody>
</table>
Display control command group

Display commands can be found in this section as well as the sections of related components. These commands control general oscilloscope settings, such as the intensity of the graticule, stacked or overlay display mode, and the fastacq color palette. Display commands also control how and where waveforms are shown, their position on screen, and zoom settings applied to the view. For example, display commands can turn on or off the display of channels or set the selected source.

Some actions can create a new view which can have its own settings. For example, adding a histogram will create a new view where the Histogram plot is displayed. Each view acts as a separate window within the oscilloscope application and can be rearranged or annotated as desired.

A WaveView is the primary view used for viewing inputs and time-domain signals. Buses, non-FFT maths, refs, analog and digital channels, and time-trends are displayed in the WaveView.

A PlotView is used for viewing measurement results and other plotted data. Histograms, eye diagrams, XY/XYZ plots, FFTs, and other plots are shown in individual PlotViews.

Each of these views can have separate settings, zoom, cursors, and annotations. Display commands which are view specific have a view parameter in the programmable interface, such as DISPLAY:WaveView<x>: ... or DISPLAY:PlotView<x>: ... Selected source can also be specified on a per-view basis. The overall selected source is determined by the selected view and the selected source within that view.

PlotViews and WaveViews have some differences in command syntax due to differences in view functionality. For example, WaveViews and PlotViews have a different zoom model. The commands for specifying the zoom reflect these differences, and different command syntax is available depending on the view.

### Table 2-25: Display control commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY?</td>
<td>Returns current display settings.</td>
</tr>
<tr>
<td>DISPLAY:COLORs</td>
<td>Sets or queries the color mode for the graticule and waveform display.</td>
</tr>
<tr>
<td>DISPLAY:GLOBAL:B&lt;x&gt;:STATE</td>
<td>Sets or queries the display mode (on or off) of the specified bus.</td>
</tr>
<tr>
<td>DISPLAY:GLOBAL:CH&lt;x&gt;:STATE</td>
<td>Sets or queries the display mode (on or off) of the specified channel (both analog and digital).</td>
</tr>
<tr>
<td>DISPLAY:GLOBAL:MATH&lt;x&gt;:STATE</td>
<td>Sets or queries the display mode (on or off) of the specified math.</td>
</tr>
<tr>
<td>DISPLAY:GLOBAL:PLOT&lt;x&gt;:STATE</td>
<td>Sets or queries the display mode (on or off) of the specified time trend plot.</td>
</tr>
<tr>
<td>DISPLAY:GLOBAL:REF&lt;x&gt;:STATE</td>
<td>Sets or queries the display mode (on or off) of the specified reference.</td>
</tr>
<tr>
<td>DISPLAY:INTENSITY?</td>
<td>Returns the waveform and graticule saturation levels.</td>
</tr>
<tr>
<td>DISPLAY:INTENSITY:BACKLight</td>
<td>Sets or queries the waveform backlight intensity settings.</td>
</tr>
</tbody>
</table>
### Table 2-25: Display control commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPlay:INTENSITy:BACKLight:AUTODim: ENABLE</td>
<td>Sets or queries the state of the display auto-dim feature.</td>
</tr>
<tr>
<td>DISPlay:INTENSITy:BACKLight:AUTODim: TIMe</td>
<td>Sets or queries the amount of time, in minutes, to wait for no user interface activity before automatically dimming the display.</td>
</tr>
<tr>
<td>DISPlay:MATHFFTView&lt;x&gt;:AUTOScale</td>
<td>Sets or returns the enabled state of autoscale for plots.</td>
</tr>
<tr>
<td>DISPlay:MATHFFTView&lt;x&gt;:GRIDlines</td>
<td>Sets or queries the grid lines setting of the plot.</td>
</tr>
<tr>
<td>DISPlay:MATHFFTView&lt;x&gt;:MATH:MATH&lt;x&gt;: STATE</td>
<td>Sets or queries the state of the specified math waveform in the specified waveview.</td>
</tr>
<tr>
<td>DISPlay:MATHFFTView&lt;x&gt;:XAXIS:SCALE</td>
<td>Sets or queries the x-axis scale setting for FFT Math waveforms.</td>
</tr>
<tr>
<td>DISPlay:MATHFFTView&lt;x&gt;:YAXIS:SCALE</td>
<td>Sets or queries the vertical scale setting for FFT Maths.</td>
</tr>
<tr>
<td>DISPlay:PERSistence</td>
<td>Sets or queries display persistence setting.</td>
</tr>
<tr>
<td>DISPlay:PERSistence:RESET</td>
<td>Clears the persistence data.</td>
</tr>
<tr>
<td>DISPlay:PLOTview&lt;x&gt;:AUTOScale</td>
<td>Sets or queries the enabled state of autoscale for plots.</td>
</tr>
<tr>
<td>DISPlay:PLOTview&lt;x&gt;:GRIDlines</td>
<td>Sets or queries the Grid lines setting of the specified plot.</td>
</tr>
<tr>
<td>DISPLAY:PLOTVIEW&lt;x&gt;:XAXIS:SCALE</td>
<td>Sets or queries the horizontal scale setting for applicable plot.</td>
</tr>
<tr>
<td>DISPLAY:PLOTVIEW&lt;x&gt;:YAXIS:SCALE</td>
<td>Sets or queries the vertical scale setting for applicable plots.</td>
</tr>
<tr>
<td>DISPlay:REFFFTView&lt;x&gt;:AUTOScale</td>
<td>Sets or queries the enabled state of auto-scale for plots.</td>
</tr>
<tr>
<td>DISPlay:REFFFTView&lt;x&gt;:GRIDlines</td>
<td>Sets or returns the grid lines setting of the plot.</td>
</tr>
<tr>
<td>DISPlay:REFFFTView&lt;x&gt;:REF:REF&lt;x&gt;: STATE</td>
<td>Sets or queries the state of the specified reference waveform in the specified waveview.</td>
</tr>
<tr>
<td>DISPlay:REFFFTView&lt;x&gt;:XAXIS:SCALE</td>
<td>Sets or queries the x-axis scale setting for REFFFT.</td>
</tr>
<tr>
<td>DISplay:SELection:BUS</td>
<td>Sets or queries the overall selected bus.</td>
</tr>
<tr>
<td>DISplay:SELection:MATH</td>
<td>Sets or queries the overall selected math.</td>
</tr>
<tr>
<td>DISplay:SELection:REFerence</td>
<td>Sets or queries the overall selected reference waveform.</td>
</tr>
<tr>
<td>DISplay:SELection:SOUrce</td>
<td>Sets or queries the overall selected source.</td>
</tr>
<tr>
<td>DISplay:SELection:VIEW</td>
<td>Sets or queries the selected view.</td>
</tr>
<tr>
<td>DISplay:SELection:WAVEView&lt;x&gt;:SOUrce</td>
<td>Sets or queries the selected source in the given waveview.</td>
</tr>
<tr>
<td>DISplay:VARpersist</td>
<td>Sets or queries the persistence decay time.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:BUS:B&lt;x&gt;:STATE</td>
<td>Sets or queries the state of the specified bus in the specified waveview.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:BUS:B&lt;x&gt;:VERTical: POSition</td>
<td>Sets or queries the vertical position of the specified bus in the specified waveview.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:CH&lt;x&gt;:STATE</td>
<td>Sets or queries the state of the specified channel in the specified waveview.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:CH&lt;x&gt;:VERTical: POSition</td>
<td>Sets or queries the vertical position of the specified channel in the specified waveview.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:CH&lt;x&gt;:VERTical: SCALE</td>
<td>Sets or queries the vertical scale of the specified channel in volts per division within the specified waveview.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:CH&lt;x&gt;_DALL:STATE</td>
<td>Sets or queries the display state of the specified digital channels in the specified waveview.</td>
</tr>
</tbody>
</table>
### Table 2-25: Display control commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:CH&lt;y&gt;_DALL:VERTical:POSition</code></td>
<td>Sets or queries the vertical position of the specified digital channel in the specified waveview in divisions.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:CH&lt;y&gt;_DALL:STATE</code></td>
<td>Sets or queries the display state of the specified digital channel in the specified waveview.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:FiLTer</code></td>
<td>Sets or queries the type of interpolation filter for the display.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:GRAticule</code></td>
<td>Selects or queries the type of graticule that is displayed.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:INTENSITy:GRAticule</code></td>
<td>Sets or queries the graticule saturation level.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:INTENSITy:WAVEform</code></td>
<td>Sets or queries the waveform saturation level.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:MATH:MATH&lt;y&gt;:AUTOScale</code></td>
<td>Sets or queries whether the specified math gets auto-scaled when the math equation changes within the specified waveview.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:MATH:MATH&lt;y&gt;:STATE</code></td>
<td>Sets or queries the state of the specified math waveform in the specified waveview.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:MATH:MATH&lt;y&gt;:VERTical:POSition</code></td>
<td>Sets or queries the vertical position in divisions of the specified math waveform.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:MATH:MATH&lt;y&gt;:VERTical:SCALE</code></td>
<td>Sets or queries the vertical scale of the specified math waveform.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:PLOT:PLOT&lt;y&gt;:AUTOScale</code></td>
<td>Sets or queries whether the specified math gets auto-scaled when the math equation changes within the specified waveview.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:PLOT:PLOT&lt;y&gt;:STATE</code></td>
<td>Sets or queries the state of the specified trend waveform in the specified waveview.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:PLOT:PLOT&lt;y&gt;:VERTical:POSition</code></td>
<td>Sets or queries the vertical position of the specified time trend in the specified waveview in divisions.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:PLOT:PLOT&lt;y&gt;:VERTical:SCALE</code></td>
<td>Sets or queries the vertical scale of the specified time trend in volts per division in the specified waveview.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;y&gt;:REF&lt;x&gt;_DALL:FRAME</code></td>
<td>Sets or returns the selected frame of the specified digital ref.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;y&gt;:REF:REF&lt;x&gt;:FRAME</code></td>
<td>Sets or returns the selected frame of the specified analog ref.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;y&gt;:REF:REF&lt;x&gt;:STATE</code></td>
<td>Sets or queries the state of the specified reference waveform in the specified waveview.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:REF:REF&lt;x&gt;:VERTical:POSition</code></td>
<td>Sets or queries the vertical position in divisions of the specified reference waveform.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:REF:REF&lt;x&gt;:VERTical:SCALE</code></td>
<td>Sets or queries the vertical scale of the specified reference waveform.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:STYle</code></td>
<td>Sets or queries the waveforms are displayed for analysis mode.</td>
</tr>
<tr>
<td><code>DISplay:WAVEView&lt;x&gt;:VIEWStyle</code></td>
<td>Sets or queries the waveform layout style used by the display.</td>
</tr>
<tr>
<td><code>DISPLAY:PLOTVIEW&lt;x&gt;:XAXIS:SCALE</code></td>
<td>Sets or queries the horizontal scale setting for applicable plots, either Linear or Log.</td>
</tr>
<tr>
<td><code>DISPLAY:PLOTVIEW&lt;x&gt;:YAXIS:SCALE</code></td>
<td>Sets or queries the vertical scale setting for applicable plots, either Linear or Log.</td>
</tr>
</tbody>
</table>
DVM Command Group

Use the commands in the DVM command group for Digital Voltmeter functionality. Requires DVM option (free with product registration).

Table 2-26: DVM Command Group

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVM</td>
<td>Resets the Digital Voltmeter measurements and history.</td>
</tr>
<tr>
<td>DVM:AUTORange</td>
<td>Sets or queries the autorange state for the Digital Voltmeter.</td>
</tr>
<tr>
<td>DVM:MEASUrement:HIStory:MAXimum?</td>
<td>Returns the maximum readout value for the DVM function over the history period.</td>
</tr>
<tr>
<td>DVM:MEASUrement:HIStory:MINimum?</td>
<td>Returns the minimum readout value for the DVM function over the history period.</td>
</tr>
<tr>
<td>DVM:MEASUrement:INFMAXimum?</td>
<td>Returns the maximum DVM readout value over the entire time that the DVM has been on since the last change using the DVM:MODE or DVM:SOURce commands or DVM RESET.</td>
</tr>
<tr>
<td>DVM:MEASUrement:INFMINimum?</td>
<td>Returns the minimum readout value of the DVM function over the entire time that the DVM has been on since the last change using the DVM:MODE or DVM:SOURce commands or DVM RESET.</td>
</tr>
<tr>
<td>DVM:MEASUrement:VALUE?</td>
<td>Returns the DVM readout value.</td>
</tr>
<tr>
<td>DVM:MODE</td>
<td>Specifies or queries the mode to use for the Digital Voltmeter.</td>
</tr>
<tr>
<td>DVM:SOURce</td>
<td>Sets or queries the source for the Digital Voltmeter.</td>
</tr>
<tr>
<td>DVM:TRIGger:FREQuency:COUNTer</td>
<td>Sets or queries the state of the trigger frequency counter.</td>
</tr>
</tbody>
</table>

Ethernet Command Group

Use the commands in the Ethernet Command Group to set up the 10BASE-T, 100BASE-TX, 1000BASE-TX or 100BASE-T Ethernet remote interface.

Table 2-27: Ethernet Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETHERnet:DHCPbootp</td>
<td>Specifies the network initialization search for a DHCP/BOOTP server.</td>
</tr>
<tr>
<td>ETHERnet:DNS:IPADDress</td>
<td>Specifies the network Domain Name Server (DNS) IP address.</td>
</tr>
<tr>
<td>ETHERnet:DOMAINname</td>
<td>Specifies the network domain name.</td>
</tr>
<tr>
<td>ETHERnet:ENET:ADDRESS?</td>
<td>Returns the Ethernet address (MAC address) value assigned to the oscilloscope.</td>
</tr>
<tr>
<td>ETHERnet:GATEWay:IPADDress</td>
<td>Specifies the network gateway IP address.</td>
</tr>
<tr>
<td>ETHERnet:IPADDress</td>
<td>Specifies the IP address assigned to the oscilloscope.</td>
</tr>
<tr>
<td>ETHERnet:LXI:LAN:RESET</td>
<td>Resets the LXI local area network.</td>
</tr>
<tr>
<td>ETHERnet:LXI:LAN:SERVICENAME</td>
<td>Specifies the mDNS service name used for the LXI interface.</td>
</tr>
</tbody>
</table>
### Table 2-27: Ethernet Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETHERnet:NAME</td>
<td>Sets or queries the instrument Ethernet hostname assigned to the oscilloscope.</td>
</tr>
<tr>
<td>ETHERnet:NETWORKCONFig</td>
<td>Specifies the Ethernet network configuration setting.</td>
</tr>
<tr>
<td>ETHERnet:PING</td>
<td>Causes the oscilloscope to ping the gateway IP address.</td>
</tr>
<tr>
<td>ETHERnet:PING:STATus?</td>
<td>Returns the results from sending the ETHERnet:PING command to ping the gateway IP address.</td>
</tr>
<tr>
<td>ETHERnet:SUBNETMask</td>
<td>Specifies the network subnet mask value.</td>
</tr>
</tbody>
</table>
**File system command group**

Use the commands in the File System Command Group to help you use the built-in hard disk drive. You can use the commands to do the following:

- List the contents of the current directory
- Create and delete directories
- Create, copy, read, rename, or delete a file

When using these commands, keep the following points in mind:

- File arguments are always enclosed within double quotes: 
  “C:\MYDIR\TEK00001.SET”
- File names follow the MS-DOS format: [DRIVE:]\[PATH\]filename
- Path separators can be either forward slashes (/) or back slashes (\)

**NOTE.** Using a back slash as a path separator can produce some unexpected results, depending on how your controller application treats escaped characters. Many applications recognize the sequence of a back slash followed by an alphabetic character as an escaped character, and, as such, interpret that alphabetic character as a control character. For example, the sequence “\n” might be interpreted as a newline character; “\t” might be interpreted as a tab character. To ensure that this interpretation does not occur, you can use double back slashes. For example, “C:\testfile.txt”.

- Some FILESystem commands can fail because a file has read-only attributes. You will not be able to delete or replace such files until this attribute is removed. Refer to the operating system help on file properties for further information.

<table>
<thead>
<tr>
<th>Table 2-28: File system commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
</tr>
<tr>
<td>FILESystem?</td>
</tr>
<tr>
<td>FILESystem:COPY</td>
</tr>
<tr>
<td>FILESystem:CWD</td>
</tr>
<tr>
<td>FILESystem:DELETE</td>
</tr>
<tr>
<td>FILESystem:DIR?</td>
</tr>
<tr>
<td>FILESystem:HOMEDir?</td>
</tr>
<tr>
<td>FILESystem:LDIR?</td>
</tr>
<tr>
<td>FILESystem:MKDir</td>
</tr>
<tr>
<td>FILESystem:READFile</td>
</tr>
<tr>
<td>FILESystem:REName</td>
</tr>
<tr>
<td>Command</td>
</tr>
<tr>
<td>----------------------</td>
</tr>
<tr>
<td>FILESyst:RMDir</td>
</tr>
<tr>
<td>FILESyst:UNMOUNT:DRive</td>
</tr>
<tr>
<td>FILESyst:WRITEFile</td>
</tr>
</tbody>
</table>
Horizontal command group

Horizontal commands control the time base of the instrument. You can set the time per division (or time per point) of the main time base. You can use the Horizontal commands to do the following:

- Set the scale, horizontal position and reference, and units of the time base
- Get the screen resolution, time of first point and time of last point, or get all the horizontal settings
- Enable or disable the display of the time base

Table 2-29: Horizontal commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQuire:NUMFRAMESACQuired?</td>
<td>Returns the number of FastFrame frames which have been acquired.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:SCALERATio</td>
<td>Sets or returns the scale ratio for the specified analog channel.</td>
</tr>
<tr>
<td>HORizontal?</td>
<td>Queries the current horizontal settings.</td>
</tr>
<tr>
<td>HORizontal:ACQDURATION?</td>
<td>Returns the time base duration.</td>
</tr>
<tr>
<td>HORizontal:DELAY:MODE</td>
<td>Sets or queries the horizontal delay mode.</td>
</tr>
<tr>
<td>HORizontal:DELAY:TIME</td>
<td>Sets or queries the horizontal delay time (position) that is used when delay is on.</td>
</tr>
<tr>
<td>HORizontal:DIVisions?</td>
<td>Returns the number of graticule divisions over which the waveform is displayed.</td>
</tr>
<tr>
<td>HORizontal:FASTframe?</td>
<td>Returns all information under horizontal:fastframe.</td>
</tr>
<tr>
<td>HORizontal:FASTframe:COUNt</td>
<td>Sets or returns the number of frames.</td>
</tr>
<tr>
<td>HORizontal:FASTframe:MAXFRames?</td>
<td>Returns the maximum number of frames.</td>
</tr>
<tr>
<td>HORizontal:FASTframe:MULtipleframes:MODE</td>
<td>Sets or returns the overlay display type.</td>
</tr>
<tr>
<td>HORizontal:FASTframe:REF:FRame</td>
<td>Sets or returns the reference frame number.</td>
</tr>
<tr>
<td>HORizontal:FASTframe:REF:INCLUDe</td>
<td>Sets or returns whether the reference frame delta information is shown in the display.</td>
</tr>
<tr>
<td>HORizontal:FASTframe:SELECTED</td>
<td>Sets or returns the selected frame number for acquired frames.</td>
</tr>
<tr>
<td>HORizontal:FASTframe:STATE</td>
<td>Sets or returns the state of FastFrame.</td>
</tr>
<tr>
<td>HORizontal:FASTframe:SUMFrame?</td>
<td>Sets or returns the summary frame type.</td>
</tr>
<tr>
<td>HORizontal:FASTframe:SUMFrame:STATE</td>
<td>Sets or returns the state of FastFrame summary frame.</td>
</tr>
<tr>
<td>HORizontal:FASTframe:TIMESTAMP:DELTA?</td>
<td>Returns the time difference between the Selected and Reference time-stamps.</td>
</tr>
<tr>
<td>HORizontal:FASTframe:TIMESTAMP:ALL?</td>
<td>Returns the time stamp of all frames.</td>
</tr>
<tr>
<td>HORizontal:FASTframe:XZERo:ALL?</td>
<td>Returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DAta:SOUrce command for all frames.</td>
</tr>
</tbody>
</table>
Table 2-29: Horizontal commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HORizontal:FASTframe:XZEro:REF?</strong></td>
<td>Returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATA:SOURce command for the reference frame.</td>
</tr>
<tr>
<td><strong>HORizontal:FASTframe:XZEro:SELECTED?</strong></td>
<td>Returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATA:SOURce command for the selected frame.</td>
</tr>
<tr>
<td><strong>HORizontal:MAIN:INTERPRatio?</strong></td>
<td>Returns the main horizontal time base interpolation ratio.</td>
</tr>
<tr>
<td><strong>HORizontal:MODE</strong></td>
<td>Sets or queries the horizontal mode.</td>
</tr>
<tr>
<td><strong>HORizontal:MODE:Automatic:FASTAcq:RECOrdlength:MAXimum:VALUE</strong></td>
<td>Sets or queries the horizontal FastAcq maximum record length.</td>
</tr>
<tr>
<td><strong>HORizontal:MODE:Automatic:FASTAcq:RECOrdlength:MAXimum:ZOOMOVERride</strong></td>
<td>Sets or queries the flag which allows override of the horizontal FastAcq maximum record length.</td>
</tr>
<tr>
<td><strong>HORizontal:MODE:MANual:CONFIGure</strong></td>
<td>Sets or queries which horizontal control (scale or record length) will also change when the sample rate is adjusted.</td>
</tr>
<tr>
<td><strong>HORizontal:MODE:RECOrdlength</strong></td>
<td>Sets or queries the record length.</td>
</tr>
<tr>
<td><strong>HORizontal:MODE:SAMPLERate</strong></td>
<td>Sets or queries the sample rate.</td>
</tr>
<tr>
<td><strong>HORizontal:MODE:SCAle</strong></td>
<td>Sets or queries the horizontal scale.</td>
</tr>
<tr>
<td><strong>HORizontal:POSITION</strong></td>
<td>Sets or queries the waveform horizontal position, in percent, that is used when delay is off.</td>
</tr>
<tr>
<td><strong>HORizontal:PREViewstate?</strong></td>
<td>Returns the display system preview state.</td>
</tr>
<tr>
<td><strong>HORizontal:RECOrdlength</strong></td>
<td>Sets or queries the horizontal record length.</td>
</tr>
<tr>
<td><strong>HORizontal:ROLL?</strong></td>
<td>Queries the horizontal roll mode status.</td>
</tr>
<tr>
<td><strong>HORizontal:SAMPLERate</strong></td>
<td>Sets or queries the horizontal sample rate.</td>
</tr>
<tr>
<td><strong>HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERride</strong></td>
<td>Sets or queries the flag which allows override of the horizontal analyze minimum sample rate.</td>
</tr>
<tr>
<td><strong>HORizontal:SAMPLERate:ANALYZemode:MINimum:VALUE</strong></td>
<td>Sets or queries the minimum sample rate used by Analysis Automatic horizontal mode.</td>
</tr>
<tr>
<td><strong>HORizontal:SCAle</strong></td>
<td>Sets or queries the horizontal scale.</td>
</tr>
</tbody>
</table>
**Command groups**

**Mask command group**

Mask commands compare incoming waveforms to standard or user-defined masks. A mask is a set of polygonal regions on the screen. Unlike limit testing, the inside of a mask is the region where waveform data would not normally fall. Standards with eye patterns usually have three masks, but some have four.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EYEMASK:MASK&lt;x&gt;:COUNT:HITS?</td>
<td>Returns the total number of hit violations for all segments in the specified eye diagram mask test.</td>
</tr>
<tr>
<td>EYEMASK:MASK&lt;x&gt;:COUNT:SEG&lt;y&gt;:HITS?</td>
<td>Returns the number of hit violations for the specified segment (area) in the specified eye diagram mask test.</td>
</tr>
<tr>
<td>EYEMASK:MASK&lt;x&gt;:CREATor?</td>
<td>Returns the name of the eye diagram plot that created the mask.</td>
</tr>
<tr>
<td>EYEMASK:MASK&lt;x&gt;:ENAbled</td>
<td>Enables or disables eye mask testing in the specified plot.</td>
</tr>
<tr>
<td>EYEMASK:MASK&lt;x&gt;:MASKfile</td>
<td>Sets or queries the current mask definition file name for the specified mask test.</td>
</tr>
<tr>
<td>EYEMASK:MASK&lt;x&gt;:TEST:SAMple:THReshold</td>
<td>Sets or queries the total number of hit violations that will cause a mask test failure.</td>
</tr>
</tbody>
</table>
**Math command group**

Use the commands in the Math Command Group to create and define math waveforms. Use the available math functions to define your math waveform.

The math waveform you create depends on sources listed in the math expression. If you change these sources, the math waveforms you previously defined will be affected.

Math expressions can be simple, containing no mathematical computation, such as CH1, which specifies that a waveform shows the signal source of channel 1. Math expressions can also be complex, consisting of 100 plus characters and comprising many sources, functions, and operands.

The acquisition of a live waveform can stop for several reasons: You can turn off the channel, stop the waveform (via Run/Stop from the Horiz/Acq menu), or stop the trigger (via Run/Stop from the Trig menu). When you turn off the channel, math continues and data is acquired but is not displayed. When you stop either the waveform or the trigger, the math calculation stops, and the last math calculation performed is displayed.

When a live waveform updates or a reference waveform is altered, math waveforms containing those waveforms as sources are also updated to reflect the changes. Also, sources must exist but do not need to be displayed to be used in and to update math waveforms.

*NOTE.* Math commands are present once a math has been added.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISplay:GLObal:MATH&lt;x&gt;:STATE</td>
<td>Sets or queries the global state of the specified math.</td>
</tr>
<tr>
<td>DISplay:SELect:MATH</td>
<td>Sets or queries the overall selected math.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:MATH:MATH&lt;x&gt;:STATE</td>
<td>Sets or queries the state of the specified math waveform in the specified waveview.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:MATH:MATH&lt;x&gt;:VERTical:POSition</td>
<td>Sets or queries the vertical position in divisions of the specified math waveform.</td>
</tr>
<tr>
<td>DISplay:WAVEView&lt;x&gt;:MATH:MATH&lt;x&gt;:VERTical:SCAle</td>
<td>Sets or queries the vertical scale of the specified math waveform.</td>
</tr>
<tr>
<td>MATH:ADDNew</td>
<td>Adds the specified math.</td>
</tr>
<tr>
<td>MATHArbfil&lt;x&gt;:FILepath</td>
<td>Sets the file path for a file of filter coefficients and reads the file.</td>
</tr>
<tr>
<td>MATH:DELeTe</td>
<td>Deletes the specified math.</td>
</tr>
<tr>
<td>MATH:LIST?</td>
<td>Lists all currently defined math waveforms.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:AVG:MODE</td>
<td>Sets or queries the math average mode flag.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:AVG:WEIght</td>
<td>Sets or queries the number of acquisitions at which the averaging algorithm will begin exponential averaging.</td>
</tr>
</tbody>
</table>
Table 2-31: Math commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH:MATH&lt;x&gt;:FUNCTION</td>
<td>Sets or queries the basic math arithmetic function.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:LABel:COlor</td>
<td>Sets or queries color of the specified math's label.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:LABel:FONT:BOLD</td>
<td>Sets or queries bold state of the specified math label.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:LABel:FONT:ITALic</td>
<td>Sets or queries italic state of the specified math label.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:LABel:FONT:SIZE</td>
<td>Sets or queries font size of the specified math label.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:LABel:FONT:TYPE</td>
<td>Sets or queries font type of the specified math label.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:LABel:FONT:UNDERline</td>
<td>Sets or queries the underline state of the specified math label.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:LABel:XPOS</td>
<td>Sets or queries the X screen offset where the math waveform label is displayed.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:LABel:YPOS</td>
<td>Sets or queries the Y screen offset where the math waveform label is displayed.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:DEFine</td>
<td>Defines new waveforms using mathematical expressions.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:GATing</td>
<td>Specifies or returns the gating setting.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:LABel:NAME</td>
<td>Sets or queries the label string.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:SOUrce&lt;x&gt;</td>
<td>Sets or queries the specified math source.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:SPECTral:PHASE</td>
<td>Sets or queries the units of a SpectralPhase function in the specified math definition string.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:SPECTral:SOUrce</td>
<td>Sets or queries the specified spectral math source.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:SPECTral:SUPPress</td>
<td>Sets or queries whether suppression threshold for the specified math waveform is enabled.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:SPECTral:SUPPress:VALue</td>
<td>Sets or queries in volts the value of suppression threshold of the specified math waveform.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:SPECTral:TYPE</td>
<td>Sets or queries the FFT type selected for spectral analysis.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:SPECTral:UNWRap</td>
<td>Sets or queries whether phase unwrap of the spectral analyzer output data is enabled.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:SPECTral:UNWRap:DEGrees</td>
<td>Sets or queries in degrees the value of unwrap phase.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:SPECTral:WINdow</td>
<td>Sets or queries the window function used to multiply the spectral analyzer input data for the specified math waveform.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:TYPe</td>
<td>Sets or queries the math type.</td>
</tr>
<tr>
<td>MATH:MATH&lt;x&gt;:VUNIT</td>
<td>Sets or queries the math custom vertical units.</td>
</tr>
</tbody>
</table>
Use the commands in the Measurement Command Group to control the automated measurement system.

Measurement commands can set and query measurement parameters. You can assign parameters, such as waveform sources and reference levels, differently for each measurement.

Clock recovery, edge, filter, gating, population and range measurement commands can be either global or per-measurement.

Global clock recovery commands are of the form
:MEASUREMENT:CLOCKRecovery:XXX

Global edge commands are of the form
:MEASUREMENT:XXXX

Global filter commands are of the form
:MEASUREMENT:FILTERs:XXX

Global gating commands are of the form
:MEASUREMENT:GATING:XXXX

Global population commands are of the form
:MEASUREMENT:POPULATION:XXX

Global range commands are of the form
:MEASUREMENT:MEASRange:XXX

Reference levels for measurements can be global, per-measurement or per-source. The default is global. Per-measurement settings are used when local reference levels are enabled for the measurement with the command
:MEASUREMENT:MEAS?:GLOBALref 0

Per-source settings are used when per-source settings are enabled with the command
:MEASUREMENT:REFLevels:TYPE PerSource

and per-source settings are selected for the measurement with the command
:MEASUREMENT:MEAS?:GLOBALref 0.

Global reference level commands are of the form
:MEASUREMENT:REFLevels:XXX

Per-source reference level commands are of the form
:MEASUREMENT:CH<x>:REFLevels:XXX

:MEASUREMENT:MATH1:REFLevels:XXX

---

### Table 2-32: Measurement commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASurement?</td>
<td>This command returns all measurement parameters.</td>
</tr>
<tr>
<td>MEASurement:ADDMEAS</td>
<td>This command adds a measurement.</td>
</tr>
<tr>
<td>MEASurement:ADDNew</td>
<td>Adds the specified measurement.</td>
</tr>
<tr>
<td>MEASurement:ANNOTate</td>
<td>This command sets or queries the annotation state for measurements.</td>
</tr>
<tr>
<td>MEASurement:AUTOset</td>
<td>Performs an analysis jitter autoset.</td>
</tr>
<tr>
<td>MEASurement:CH&lt;x&gt;:REFLevels:ABSolute: FALLHigh</td>
<td>This command sets or queries the value used as the high reference level of the falling edge when the source ref level method is set to absolute.</td>
</tr>
<tr>
<td>MEASurement:CH&lt;x&gt;:REFLevels:ABSolute: FALLLow</td>
<td>This command sets or queries the value used as the low reference level of the falling edge when the source ref level method is set to absolute.</td>
</tr>
</tbody>
</table>
Table 2-32: Measurement commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:ABSolute:FALLMid</td>
<td>This command sets or queries the value used as the mid reference level of the falling edge when the source ref level method is set to absolute.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:ABSolute:HYSTeresis</td>
<td>This command sets or queries the value of the hysteresis of the reference level when the source ref level method is set to absolute.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:ABSolute:RISEHigh</td>
<td>This command sets or queries the value used as the high reference level of the rising edge when the source ref level method is set to absolute.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:ABSolute:RISELow</td>
<td>This command sets or queries the value used as the low reference level of the rising edge when the source ref level method is set to absolute.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:ABSolute:RISEMId</td>
<td>This command sets or queries the value used as the mid reference level of the rising edge when the source ref level method is set to absolute.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:ABSolute:TYPE</td>
<td>This command sets or queries the reference level type for the source.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:BASETop</td>
<td>This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:METHod</td>
<td>This command sets or queries the method used to calculate reference levels for the source.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:PERCent:FALLHigh</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the source ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:PERCent:FALLLow</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:PERCent:FALLMid</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:PERCent:HYSTeresis</td>
<td>This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the source ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:PERCent:RISEHigh</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the source ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:PERCent:RISELow</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the source ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:PERCent:RISEMId</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the source ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUREment:CH&lt;x&gt;:REFLevels:PERCent:TYPE</td>
<td>This command specifies or queries the reference level percent type for the source.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:ADVanced:METHOD</td>
<td>This command sets or queries the global advanced clock recovery method.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:CLOCKFrequency</td>
<td>This command sets or queries the global clock frequency used for fixed constant clock recovery.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:CLOCKMultiplier</td>
<td>This command sets or queries the global clock multiplier used for explicit clock recovery.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:CONSTCLOCKMODe</td>
<td>This command sets or queries the global constant clock mode used for constant clock recovery.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:DAMPing</td>
<td>This command sets or queries the global damping value used for PLL clock recovery.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:DATAPath</td>
<td>This command sets or queries the global file containing the data pattern used for known data pattern clock recovery.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:DATARate</td>
<td>This command sets or queries the global nominal data bit rate used for nominal data rate clock recovery.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:EXPLICITCLOCKMODe</td>
<td>This command sets or queries the global explicit clock mode used for explicit clock recovery.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:JTFBandwidth</td>
<td>This command sets or queries the global JTF bandwidth used for PLL clock recovery.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:LOOPBandwidth</td>
<td>This command sets or queries the global loop bandwidth used for PLL clock recovery.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:MEANAUTOCalculate</td>
<td>This command sets or queries how often the clock is calculated for constant clock recovery.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:METHod</td>
<td>This command sets or queries the global clock recovery method.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:MODel</td>
<td>This command sets or queries the global PLL clock recovery model used for PLL clock recovery.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:NOMINALOFFset</td>
<td>This command sets or queries the global offset value used for explicit clock recovery.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:NOMINALOFFset:SELECTIONtype</td>
<td>This command sets or queries the global offset type used for explicit clock recovery.</td>
</tr>
<tr>
<td>MEASUrement:CLOCKRecovery:STAndard</td>
<td>This command sets or queries the global communications standard used for PLL clock recovery.</td>
</tr>
<tr>
<td>MEASUrement:DELeTe</td>
<td>The command deletes the specified measurement.</td>
</tr>
<tr>
<td>MEASUrement:DIRacmodel</td>
<td>This command sets or queries the dirac model used to separate random from deterministic jitter for jitter measurements.</td>
</tr>
<tr>
<td>MEASUrement:DISPLAYUnits</td>
<td>This command sets or queries the display units used for jitter summary measurements.</td>
</tr>
<tr>
<td>MEASUrement:EDGE&lt;x&gt;</td>
<td>Sets or queries the type of the edge for the measurement.</td>
</tr>
<tr>
<td>MEASUrement:EYERENDER</td>
<td>This command sets or queries the state of high-performance eye rendering for an eye diagram.</td>
</tr>
<tr>
<td>MEASUrement:FILTers:BLANKingtime</td>
<td>This command sets or queries the global filter blanking time.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>MEASUREMENT:FILTers:HIGHPass:FREQ</td>
<td>This command sets or queries the global high pass filter frequency.</td>
</tr>
<tr>
<td>MEASUREMENT:FILTers:HIGHPass:SPEC</td>
<td>This command sets or queries the global high pass filter order.</td>
</tr>
<tr>
<td>MEASUREMENT:FILTers:LOWPass:FREQ</td>
<td>This command sets or queries the global low pass filter cutoff frequency.</td>
</tr>
<tr>
<td>MEASUREMENT:FILTers:LOWPass:SPEC</td>
<td>This command sets or queries the global low pass filter order.</td>
</tr>
<tr>
<td>MEASUREMENT:FILTers:RAMPtime</td>
<td>This command sets or queries the global filter ramp time.</td>
</tr>
<tr>
<td>MEASUREMENT:GATING</td>
<td>This command sets or queries the global gating type.</td>
</tr>
<tr>
<td>MEASUREMENT:GATING:ACTIVE</td>
<td>This command sets or queries the global gating active level used for logic gating.</td>
</tr>
<tr>
<td>MEASUREMENT:GATING:HYSTeresis</td>
<td>This command sets or queries the global gating hysteresis value used for logic gating.</td>
</tr>
<tr>
<td>MEASUREMENT:GATING:LOGICSource</td>
<td>This command sets or queries the gating data source used for logic gating.</td>
</tr>
<tr>
<td>MEASUREMENT:GATING:MICRORef</td>
<td>This command sets or queries the global gating mid reference value used for logic gating.</td>
</tr>
<tr>
<td>MEASUREMENT:GATING:SEARCHSource</td>
<td>This command sets or queries the global gating search source used for logic gating.</td>
</tr>
<tr>
<td>MEASUREMENT:INTERp</td>
<td>This command sets or queries the interpolation mode used to locate edge crossings.</td>
</tr>
<tr>
<td>MEASUREMENT:JITTERMODEL</td>
<td>This command sets or queries the model used to separate random from deterministic jitter for jitter measurements.</td>
</tr>
<tr>
<td>MEASUREMENT:LIST?</td>
<td>Lists all currently defined measurements.</td>
</tr>
<tr>
<td>MEASUREMENT:LOCKRJ</td>
<td>Sets or queries the state of RJ locking.</td>
</tr>
<tr>
<td>MEASUREMENT:LOCKRJValue</td>
<td>Sets or queries the RJ lock value.</td>
</tr>
<tr>
<td>MEASUREMENT:MATH&lt;x&gt;:REFLevels:ABSolute:FALLHigh</td>
<td>This command sets or queries the value used as the high reference level of the falling edge when the source reference level method is set to absolute.</td>
</tr>
<tr>
<td>MEASUREMENT:MATH&lt;x&gt;:REFLevels:ABSolute:FALLLow</td>
<td>This command sets or queries the value used as the low reference level of the falling edge when the source reference level method is set to absolute.</td>
</tr>
<tr>
<td>MEASUREMENT:MATH&lt;x&gt;:REFLevels:ABSolute:FALLMid</td>
<td>This command sets or queries the value used as the mid reference level of the falling edge when the source reference level method is set to absolute.</td>
</tr>
<tr>
<td>MEASUREMENT:MATH&lt;x&gt;:REFLevels:ABSolute:HYSTeresis</td>
<td>This command sets or queries the value of the hysteresis of the reference level when the source reference level method is set to absolute.</td>
</tr>
<tr>
<td>MEASUREMENT:MATH&lt;x&gt;:REFLevels:ABSolute:RISEHigh</td>
<td>This command sets or queries the value used as the high reference level of the rising edge when the source reference level method is set to absolute.</td>
</tr>
<tr>
<td>MEASUREMENT:MATH&lt;x&gt;:REFLevels:ABSolute:RISELow</td>
<td>This command sets or queries the value used as the low reference level of the rising edge when the source reference level method is set to absolute.</td>
</tr>
<tr>
<td>MEASUREMENT:MATH&lt;x&gt;:REFLevels:ABSolute:RISEMid</td>
<td>This command sets or queries the value used as the mid reference level of the rising edge when the source reference level method is set to absolute.</td>
</tr>
<tr>
<td>MEASUREMENT:MATH&lt;x&gt;:REFLevels:ABSolute:TYPE</td>
<td>This command sets or queries the reference level type for the source.</td>
</tr>
<tr>
<td>MEASUREMENT:MATH&lt;x&gt;:REFLevels:BASETop</td>
<td>This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the source.</td>
</tr>
</tbody>
</table>
Table 2-32: Measurement commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASUrement:MATH&lt;x&gt;:REFLevels:METHod</td>
<td>This command sets or queries the method used to calculate reference levels for the source.</td>
</tr>
<tr>
<td>MEASUrement:MATH&lt;x&gt;:REFLevels:PERCent: FALLHigh</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the source ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUrement:MATH&lt;x&gt;:REFLevels:PERCent: FALLLow</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUrement:MATH&lt;x&gt;:REFLevels:PERCent: FALLMid</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUrement:MATH&lt;x&gt;:REFLevels:PERCent: HYSTeresis</td>
<td>This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the source ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUrement:MATH&lt;x&gt;:REFLevels:PERCent: RISEHigh</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the source ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUrement:MATH&lt;x&gt;:REFLevels:PERCent: RISELow</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the source ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUrement:MATH&lt;x&gt;:REFLevels:PERCent: RISEMid</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the source ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUrement:MATH&lt;x&gt;:REFLevels:PERCent: TYPE</td>
<td>This command specifies or queries the reference level percent type for the source.</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:BER</td>
<td>This command sets or queries the BER value for the measurement.</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:BER:TARGETBER</td>
<td>This command sets or queries the target BER value for the measurement.</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:BIN</td>
<td>This command sets or queries the bin count for the measurement.</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:BITCfgmode</td>
<td>This command sets or queries whether the measurement returns the mean or mode statistic result when the measurement type is bit amplitude/high/low.</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:BITEnd</td>
<td>This command sets or queries the bit end as a percentage of the unit interval.</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:BITPcnt</td>
<td>This command sets or queries the bit center as a percentage of the unit interval.</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:BITSTart</td>
<td>This command sets or queries the bit start as a percentage of the unit interval.</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:BITType</td>
<td>This command sets or queries the bit type for the measurement.</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:BURSTEDGTYPE</td>
<td>This command sets or queries the burst edge type for the measurement.</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:CCResUIts: ALLAcqs:MAXimum?</td>
<td>This query-only command returns the maximum cycle-cycle value for the specified measurement for all acquisitions.</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:CCResUIts: ALLAcqs:MEAN?</td>
<td>This query-only command returns the mean cycle-cycle value for the specified measurement for all acquisitions.</td>
</tr>
</tbody>
</table>
Table 2-32: Measurement commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CCRESUltS:ALLAcqs:MINimum?</td>
<td>This query-only command returns the minimum cycle-cycle value for the specified measurement for all acquisitions.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CCRESUltS:ALLAcqs:PK2PK?</td>
<td>This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for all acquisitions.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CCRESUltS:ALLAcqs:POPUlation?</td>
<td>This query-only command returns the population of all cycle-cycle statistics for the specified measurement for all acquisitions accumulated since statistics were last reset.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CCRESUltS:ALLAcqs:STDDev?</td>
<td>This query-only command returns the standard deviation cycle-cycle for the specified measurement for all acquisitions.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CCRESUltS:CURRentacq:MAXimum?</td>
<td>This query-only command returns the maximum cycle-cycle value for the specified measurement for the current acquisition.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CCRESUltS:CURRentacq:MEAN?</td>
<td>This query-only command returns the mean cycle-cycle value for the specified measurement for the current acquisition.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CCRESUltS:CURRentacq:MINimum?</td>
<td>This query-only command returns the minimum cycle-cycle value for the specified measurement for the current acquisition.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CCRESUltS:CURRentacq:PK2PK?</td>
<td>This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for the current acquisition.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CCRESUltS:CURRentacq:POPUlation?</td>
<td>This query-only command returns the population of the cycle-cycle statistics for the specified measurement for the current acquisition.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CCRESUltS:CURRentacq:STDDev?</td>
<td>This query-only command returns the standard deviation cycle-cycle for the specified measurement for the current acquisition.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:ADVanced:METHod</td>
<td>This command sets or queries the advanced clock recovery method when advanced clock recovery is used for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:CLOCKFrequency</td>
<td>This command sets or queries the clock frequency used when fixed constant clock recovery is used for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:CLOCKMultiplier</td>
<td>This command sets or queries the clock multiplier used when explicit clock recovery is used for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:CONSTCLOCKMODe</td>
<td>This command sets or queries the constant clock mode used when constant clock recovery is used for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:DAMPing</td>
<td>This command sets or queries the damping value used when PLL clock recovery is used for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:DATAPath</td>
<td>This command sets or queries the file containing the data pattern used when known data pattern clock recovery is used for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:DATARate</td>
<td>This command sets or queries the nominal data bit rate when nominal data rate clock recovery is used for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:EXPLICITCLOCKMODe</td>
<td>This command sets or queries the explicit clock mode used when explicit clock recovery is used for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:GLOBal</td>
<td>This command sets or queries the clock recovery settings global flag for the measurement.</td>
</tr>
</tbody>
</table>
### Table 2-32: Measurement commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:JTFFrequency</code></td>
<td>This command sets or queries the JTF frequency used when PLL clock recovery is used for the measurement.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:LOOPBandwidth</code></td>
<td>This command sets or queries the loop bandwidth used when PLL clock recovery is used for the measurement.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:MEANATOCalculate</code></td>
<td>This command sets or queries how often the clock is calculated when constant clock recovery is used for the measurement.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:METHod</code></td>
<td>This command sets or queries the clock recovery method for the measurement.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:MODE</code></td>
<td>This command sets or queries the PLL clock recovery model used when PLL clock recovery is used for the measurement.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:NOMINALOFFset</code></td>
<td>This command sets or queries the offset value used when explicit clock recovery is used for the measurement.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:NOMINALOFFset:SELECTIONtype</code></td>
<td>This command sets or queries the offset type used when explicit clock recovery is used for the measurement.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:CLOCKRecovery:STAndard</code></td>
<td>This command sets or queries the communications standard when PLL clock recovery is used for the measurement.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:COMMONMode:FILTers:STATE</code></td>
<td>This command sets or queries whether a filter is used for the measurement when the measurement type is AC common mode.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:COMMONMode:SOURCES</code></td>
<td>This command sets or queries the number of sources for the measurement when the measurement type is AC common mode.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:CYCLemode</code></td>
<td>This command sets or queries the cycle mode for the measurement.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:DELeY:EDGE&lt;x&gt;</code></td>
<td>This command sets or queries the 'to edge' type when <code>EDGE?</code> is <code>EDGE1</code> and the 'from edge' type when <code>EDGE?</code> is <code>EDG2</code>, for the measurement when the measurement type is <code>DELAY</code>.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:DISPlaystat:ENABle</code></td>
<td>Turns on and off the display of statistics in measurement badges.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:EDGE&lt;x&gt;</code></td>
<td>This command sets or queries the type of the specified edge, rise or fall, for the measurement.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:EDGEIncr</code></td>
<td>This command sets or queries the edge increment value for the measurement.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:EDGES:FROMLevel</code></td>
<td>This command sets or queries the 'from level' edge for the measurement.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:EDGES:LEVEL</code></td>
<td>This sets or queries the level type for the 'time outside level' measurement.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:EDGES:LOWERFREQuency</code></td>
<td>This command sets or queries the lower frequency for the measurement when the measurement type is phase noise. Lower frequencies are ignored.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:EDGES:N</code></td>
<td>The command sets or queries the number of accumulation cycles for the measurement when the measurement type is <code>nperiod</code>.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:EDGES:SLEWRATEMethod</code></td>
<td>This command sets or queries the slew rate method for the measurement.</td>
</tr>
<tr>
<td><code>MEASurement:MEAS&lt;x&gt;:EDGES:TOLevel</code></td>
<td>This command sets or queries the 'to level' edge for the measurement.</td>
</tr>
</tbody>
</table>
### Table 2-32: Measurement commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:EDGES:UPPERFREQUENCY</td>
<td>This command sets or queries the upper frequency for the measurement when the measurement type is phase noise. Higher frequencies are ignored.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:FILTERS:BLANKINGTIME</td>
<td>This command sets or queries the filter blanking time for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:FILTERS:GLOBAL</td>
<td>This command sets or queries the global flag for filter settings for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:FILTERS:HIGHPASS:FREQ</td>
<td>This command sets or queries the high pass filter frequency for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:FILTERS:HIGHPASS:SPEC</td>
<td>This command sets or queries the high pass filter order for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:FILTERS:LOWPASS:FREQ</td>
<td>This command sets or queries the low pass filter cutoff frequency for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:FILTERS:LOWPASS:SPEC</td>
<td>This command sets or queries the low pass filter order for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:FILTERS:RAMPTIME</td>
<td>This command sets or queries the 'from edge' ramp time for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:FROMEDGE</td>
<td>This command sets or queries the 'from edge' search direction for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:GATING</td>
<td>This command sets or queries the gating type for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:GATING:ACTIVE</td>
<td>This command sets or queries the gating active level when the gating type is logic.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:GATING:GLOBAL</td>
<td>This command sets or queries the gating settings global flag.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:GATING:HYZTERESIS</td>
<td>This command sets or queries the gating hysteresis value when the gating type is logic.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:GATING:LOGICSOURCE</td>
<td>This command sets or queries the gating data source when the gating type is logic.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:GATING:MIIDREF</td>
<td>This command sets or queries the gating mid ref value when the gating type is logic.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:GATING:SEARCHSOURCE</td>
<td>This command sets or queries the gating search source when the gating type is search.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:GLOBALREF</td>
<td>This command sets or queries the reference levels global flag for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:HIGHREFVOLTAGE</td>
<td>This command sets or queries the high reference voltage value for the 'time outside level' measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:IDLETIME</td>
<td>This command sets or queries the idle time for the measurement when the measurement type is burst width.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:JITTERSUMMARY:DCD</td>
<td>This command sets or queries whether DCD is included in the jitter summary for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:JITTERSUMMARY:DDJ</td>
<td>This command sets or queries whether DDJ is included in the jitter summary for the measurement.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:JITTERSummary: DJDD</td>
<td>This command sets or queries whether DJ-dd is included in the jitter summary for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:JITTERSummary: EYEWIDTHBER</td>
<td>This command sets or queries whether EyeWidth@BER is included in the jitter summary for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:JITTERSummary: NPJ</td>
<td>This command sets or queries whether NPJ is included in the jitter summary for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:JITTERSummary: PJ</td>
<td>This command sets or queries whether PJ is included in the jitter summary for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:JITTERSummary: RJDD</td>
<td>This command sets or queries whether RJ-dd is included in the jitter summary for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:JITTERSummary: TIE</td>
<td>This command sets or queries whether TIE is included in the jitter summary for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:JITTERSummary: TJBER</td>
<td>This command sets or queries whether TJ@BER is included in the jitter summary for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:LABel</td>
<td>This command sets or queries the label for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:LOWREFVoltage</td>
<td>This command sets or queries the low reference voltage value for the 'time outside level' measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:MAXCycle</td>
<td>Sets or queries the maximum cycle value for the DDRERRN and DDRTERRMN measurements.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:MEASRange: GLOBal</td>
<td>This command sets or queries the range settings global flag for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:MEASRange: MAX</td>
<td>This command sets or queries the range maximum value for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:MEASRange: MIN</td>
<td>This command sets or queries the range minimum value for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:MEASRange: STATE</td>
<td>This command sets or queries the range state for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:MINCycle</td>
<td>Sets or queries the minimum cycle value for the DDRERRN and DDRTERRMN measurements.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:PATTERNDETECTION</td>
<td>This command sets or queries the pattern detection type for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:PATTERNLenght</td>
<td>This command sets or queries the pattern length for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:PATTERNTYPE</td>
<td>This command sets or queries the pattern type for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:PERFREQ:EDGE</td>
<td>This command sets or queries the edge type of a Period/Frequency measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:POLarity</td>
<td>This command sets or queries the polarity for the measurement when the measurement type is burst width.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:POPUlation:GLOBal</td>
<td>This command sets or queries the population settings global flag.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:POPUlation:LIMIT: STATE</td>
<td>This command sets or queries the population limit state for the measurement.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:POPUlation:LIMIT:VALue</td>
<td>This command sets or queries the population limit value for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels:ABSolute:FALLHigh</td>
<td>This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels:x&gt;:ABSolute:FALLLow</td>
<td>This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels:x&gt;:ABSolute:FALLMid</td>
<td>This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:ABSolute:HYSTeresis</td>
<td>This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:ABSolute:RISEHigh</td>
<td>This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:ABSolute:RISELow</td>
<td>This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:ABSolute:RISEMid</td>
<td>This command sets or queries the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:ABSolute:TYPE</td>
<td>This command sets or queries the reference level type for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:BASETop</td>
<td>This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:METHod</td>
<td>This command sets or queries the method used to calculate reference levels for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:FALLHigh</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:FALLLow</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement's ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:FALLMid</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:HYSTeresis</td>
<td>This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:RISEHigh</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:RISELow</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent.</td>
</tr>
</tbody>
</table>
Table 2-32: Measurement commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:RISEMid</td>
<td>This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:TYPE</td>
<td>This command specifies or queries the reference level percent type for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:REFMode</td>
<td>This command sets or queries the reference level mode for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:REFVoltage</td>
<td>This command sets or queries the reference voltage value for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:RESULTS:ALLAcqs:MAXimum?</td>
<td>This command is identical to that described in the DPOJet programmer manual.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:RESULTS:ALLAcqs:MEAN?</td>
<td>This command is identical to that described in the DPOJet programmer manual.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:RESULTS:ALLAcqs:MINimum?</td>
<td>This command is identical to that described in the DPOJet programmer manual.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:RESULTS:ALLAcqs:PK2PK?</td>
<td>This command is identical to that described in the DPOJet programmer manual.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:RESULTS:ALLAcqs:POPULATION?</td>
<td>This command is identical to that described in the DPOJet programmer manual.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:RESULTS:ALLAcqs:STDDev?</td>
<td>This command is identical to that described in the DPOJet programmer manual.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:RESULTS:CURRentacq:MAXimum?</td>
<td>This query-only command returns the maximum value found for the specified measurement since the last statistical reset.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:RESULTS:CURRentacq:MEAN?</td>
<td>This query-only command returns the mean value for the measurement for the current acquisition.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:RESULTS:CURRentacq:MINimum?</td>
<td>This query-only command returns the minimum value found for the specified measurement since the last statistical reset.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:RESULTS:CURRentacq:PK2PK?</td>
<td>This query-only command returns the peak-to-peak value for the specified measurement for the current acquisition.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:RESULTS:CURRentacq:POPULATION?</td>
<td>This query-only command returns the population for the specified measurement for the current acquisition.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:RESULTS:CURRentacq:STDDev?</td>
<td>This query-only command returns the standard deviation for the specified measurement for all acquisitions accumulated since statistics were last reset.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:SIGNALType</td>
<td>This command sets or queries the signal type of source 1 for the measurement.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:SOURCE&lt;x&gt;</td>
<td>This command sets or queries the measurement source.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:SSC:NAMElfreq</td>
<td>This command sets or queries the user-defined frequency for the measurement when the measurement type is SSC.</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:SSC:NAMElfreq:SELECTIONtype</td>
<td>This command sets or queries the frequency detection type for the measurement when the measurement type is SSC.</td>
</tr>
</tbody>
</table>
Table 2-32: Measurement commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:TCKAVG</td>
<td>This command sets or queries the average clock period value used in DDR measurements.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:TIMINGMode</td>
<td>This command sets or queries the Timing mode for the specified DDR measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:TOEdge</td>
<td>This command sets or queries the 'to edge' type for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:TOEDGESEARCHDIRECT</td>
<td>This command sets or queries the 'to edge' search direction for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:TRANSition</td>
<td>This command sets or queries the transition edges flag for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:TYPE</td>
<td>This command sets or queries the measurement type.</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:WINDOWLENGTH</td>
<td>This command sets or queries the window length for the measurement.</td>
</tr>
<tr>
<td>MEASurement:MEASRange:MAX</td>
<td>This command sets or queries the global range maximum value.</td>
</tr>
<tr>
<td>MEASurement:MEASRange:MIN</td>
<td>This command sets or queries the global range minimum value.</td>
</tr>
<tr>
<td>MEASurement:MEASRange:STATE</td>
<td>Sets or queries the global range state.</td>
</tr>
<tr>
<td>MEASurement:MINUI</td>
<td>This command sets or queries the minimum number of unit intervals required for BUJ analysis.</td>
</tr>
<tr>
<td>MEASurement:POPULATION:LIMIT:STATE</td>
<td>This command sets or queries the global population limit state.</td>
</tr>
<tr>
<td>MEASurement:POPULATION:LIMIT:VALUE</td>
<td>This command sets or queries the global population limit value.</td>
</tr>
<tr>
<td>MEASurement:REFLevels:ABSolute:FALLHigh</td>
<td>Sets or queries the value used as the high reference level of the falling edge.</td>
</tr>
</tbody>
</table>
Table 2-32: Measurement commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASUREMENT:REFLev els:PERCent:FALLMid</td>
<td>Sets or queries the percentage used to calculate the mid reference level of the falling edge.</td>
</tr>
<tr>
<td>MEASUREMENT:REFLev els:PERCent:HYSTeresis</td>
<td>Sets or queries the percentage used to calculate the hysteresis of the reference level.</td>
</tr>
<tr>
<td>MEASUREMENT:REFLev els:PERCent:RISEHigh</td>
<td>Sets or queries the percentage used to calculate the high reference level of the rising edge.</td>
</tr>
<tr>
<td>MEASUREMENT:REFLev els:PERCent:RISEL ow</td>
<td>Sets or queries the percentage used to calculate the low reference level of the rising edge.</td>
</tr>
<tr>
<td>MEASUREMENT:REFLev els:PERCent:RISEM id</td>
<td>Sets or queries the percentage used to calculate the mid reference level of the rising edge.</td>
</tr>
<tr>
<td>MEASUREMENT:REFLev els:PERCent:TYPE</td>
<td>Sets or queries the reference level percent type.</td>
</tr>
<tr>
<td>MEASUREMENT:REFLev els:TYPE</td>
<td>This command sets or queries the shared reference level method used for sources of measurement calculations.</td>
</tr>
<tr>
<td>MEASUREMENT:REF&lt;x&gt;:REFLev els:ABSolute: FALLHigh</td>
<td>Sets or queries the value used as the high reference level of the falling edge.</td>
</tr>
<tr>
<td>MEASUREMENT:REF&lt;x&gt;:REFLev els:ABSolute: FALLLow</td>
<td>Sets or queries the value used as the low reference level of the falling edge.</td>
</tr>
<tr>
<td>MEASUREMENT:REF&lt;x&gt;:REFLev els:ABSolute: FALLMid</td>
<td>Sets or queries the value used as the mid reference level of the falling edge.</td>
</tr>
<tr>
<td>MEASUREMENT:REF&lt;x&gt;:REFLev els:ABSolute: HYSTeresis</td>
<td>Sets or queries the value of the hysteresis of the reference level.</td>
</tr>
<tr>
<td>MEASUREMENT:REF&lt;x&gt;:REFLev els:ABSolute: RISEHigh</td>
<td>Sets or queries the value used as the high reference level of the rising edge.</td>
</tr>
<tr>
<td>MEASUREMENT:REF&lt;x&gt;:REFLev els:ABSolute: RISELow</td>
<td>Sets or queries the value used as the low reference level of the rising edge.</td>
</tr>
<tr>
<td>MEASUREMENT:REF&lt;x&gt;:REFLev els:ABSolute: RISEMid</td>
<td>Sets or queries the value used as the mid reference level of the rising edge.</td>
</tr>
<tr>
<td>MEASUREMENT:REF&lt;x&gt;:REFLev els:ABSolute: TYPE</td>
<td>Sets or queries the reference level type.</td>
</tr>
<tr>
<td>MEASUREMENT:REF&lt;x&gt;:REFLev els:BASETop</td>
<td>Sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels.</td>
</tr>
<tr>
<td>MEASUREMENT:REF&lt;x&gt;:REFLev els:METHOD</td>
<td>Sets or queries the method used to calculate reference levels.</td>
</tr>
<tr>
<td>MEASUREMENT:REF&lt;x&gt;:REFLev els:PERCent: FALLHigh</td>
<td>Sets or queries the percentage used to calculate the high reference level of the falling edge.</td>
</tr>
<tr>
<td>MEASUREMENT:REF&lt;x&gt;:REFLev els:PERCent: FALLLow</td>
<td>Sets or queries the percentage used to calculate the low reference level of the falling edge.</td>
</tr>
<tr>
<td>MEASUREMENT:REF&lt;x&gt;:REFLev els:PERCent: FALLMid</td>
<td>Sets or queries the percentage used to calculate the mid reference level of the falling edge.</td>
</tr>
</tbody>
</table>
### Table 2-32: Measurement commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASUrement:REF&lt;x&gt;:REFLevels:PERCent:HYSTeresis</td>
<td>Sets or queries the percentage used to calculate the hysteresis of the reference level.</td>
</tr>
<tr>
<td>MEASUrement:REF&lt;x&gt;:REFLevels:PERCent:RISEHigh</td>
<td>Sets or queries the percentage used to calculate the high reference level of the rising edge.</td>
</tr>
<tr>
<td>MEASUrement:REF&lt;x&gt;:REFLevels:PERCent:RISELow</td>
<td>Sets or queries the percentage used to calculate the low reference level of the rising edge.</td>
</tr>
<tr>
<td>MEASUrement:REF&lt;x&gt;:REFLevels:PERCent:RISEMid</td>
<td>Sets or queries the percentage used to calculate the mid reference level of the rising edge.</td>
</tr>
<tr>
<td>MEASUrement:REF&lt;x&gt;:REFLevels:PERCent:TYPE</td>
<td>Sets or queries the reference level percent type.</td>
</tr>
<tr>
<td>MEASUrement:STATIstics:CYCLEMode</td>
<td>This command sets or queries whether cycle-cycle statistics are calculated for all measurements.</td>
</tr>
</tbody>
</table>
Miscellaneous command group

Miscellaneous commands do not fit into other categories.

Several commands and queries are common to all devices. The 488.2-1987 standard defines these commands. The common commands begin with an asterisk (*) character.

Table 2-33: Miscellaneous commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOSAVEPITIMEOUT</td>
<td>Sets or queries the idle time from the programmable interface before auto-save occurs.</td>
</tr>
<tr>
<td>AUTOSAVEUITIMEOUT</td>
<td>Sets or queries the idle time from the user interface before auto-save occurs.</td>
</tr>
<tr>
<td>AUTOSet</td>
<td>Sets or queries the vertical, horizontal, and trigger controls of the instrument to automatically acquire and display the selected waveform.</td>
</tr>
<tr>
<td>AUTOSet:ACQuisition:ENAble</td>
<td>Sets or queries the Autoset acquisition setting adjustment.</td>
</tr>
<tr>
<td>AUTOSet:ENAble</td>
<td>Sets or queries the Autoset enable/disable mode.</td>
</tr>
<tr>
<td>AUTOSet:HORizontal:ENAble</td>
<td>Sets or queries Autoset's adjustment of horizontal settings.</td>
</tr>
<tr>
<td>AUTOSet:TRIGger:ENAble</td>
<td>Sets or queries Autoset's adjustment of trigger settings.</td>
</tr>
<tr>
<td>AUTOSet:VERTical:ENAble</td>
<td>sets or queries Autoset's adjustment of vertical settings.</td>
</tr>
<tr>
<td>AUTOSet:VERTical:OPTIMize</td>
<td>Sets or queries which vertical settings Autoset will optimize when the display mode is set to Overlay mode.</td>
</tr>
<tr>
<td>AUXout:EDGE</td>
<td>Sets or queries the direction in which the trigger output signal will transition when a trigger occurs.</td>
</tr>
<tr>
<td>AUXout:SOUrce</td>
<td>Sets or queries the trigger source at the BNC connection.</td>
</tr>
<tr>
<td>CLEAR</td>
<td>Clears acquisitions, measurements, and waveforms.</td>
</tr>
<tr>
<td>DATE?</td>
<td>Queries the date that the instrument displays.</td>
</tr>
<tr>
<td>*DDT</td>
<td>Sets or queries the commands that will be executed by the group execute trigger.</td>
</tr>
<tr>
<td>FPAnele:PRESS</td>
<td>Turns off the displayed menu.</td>
</tr>
<tr>
<td>FPAnele:TURm</td>
<td>Emulates a knob turn.</td>
</tr>
<tr>
<td>HEADer</td>
<td>Sets or queries the Response Header Enable State.</td>
</tr>
<tr>
<td>ID?</td>
<td>Returns identifying information about the instrument and its firmware.</td>
</tr>
<tr>
<td>*IDN?</td>
<td>Returns the instrument identification code.</td>
</tr>
<tr>
<td>LICense?</td>
<td>Queries all license parameters.</td>
</tr>
<tr>
<td>LICENSE:APPID?</td>
<td>Returns a comma-separated list of the active application IDs.</td>
</tr>
<tr>
<td>LICENSE:COUNT?</td>
<td>Returns a count of the number of active licenses installed.</td>
</tr>
<tr>
<td>LICENSE:ERRror?</td>
<td>This query-only command prompts the instrument to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue (alias for ALLEV?).</td>
</tr>
<tr>
<td>LICENSE:GMT?</td>
<td>Returns the GMT time in ISO 8601 format, the local date, 24 hour time and time-zone offset.</td>
</tr>
<tr>
<td>LICENSE:HID?</td>
<td>Returns the instrument HostID unique identifier.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LIC:INSTall</td>
<td>Accepts a &lt;block data&gt; license and installs it on the instrument.</td>
</tr>
<tr>
<td>LIC:ITEM?</td>
<td>Returns the details pertaining to a specific license.</td>
</tr>
<tr>
<td>LIC:LIST?</td>
<td>Returns the active license nomenclatures as a comma-separated list of strings.</td>
</tr>
<tr>
<td>LIC:VALIDate?</td>
<td>Accepts a license nomenclature as an argument and returns the status of the license.</td>
</tr>
<tr>
<td>LIC:UNINSTALL?</td>
<td>Returns the exit license information for the user to return to their TekAMS account.</td>
</tr>
<tr>
<td>LOCK</td>
<td>Sets or queries the front panel lock state.</td>
</tr>
<tr>
<td>*LRN?</td>
<td>Returns a listing of instrument settings.</td>
</tr>
<tr>
<td>MAINWindow:RRBDisplaystate</td>
<td>Sets the display state of the Results readout bar to ON (displayed) or OFF (not displayed).</td>
</tr>
<tr>
<td>NEWpass</td>
<td>Changes the password for user protected data.</td>
</tr>
<tr>
<td>PASSword</td>
<td>Provides access for changing user protected data.</td>
</tr>
<tr>
<td>PAUSE</td>
<td>Causes the interface to pause the specified number of seconds before processing any other commands.</td>
</tr>
<tr>
<td>REM</td>
<td>Specifies a comment which is ignored by the instrument.</td>
</tr>
<tr>
<td>ROSc:SOURce</td>
<td>Selects or queries the selected source for the time base reference oscillator.</td>
</tr>
<tr>
<td>ROSc:STATE?</td>
<td>Returns whether the time base reference oscillator is locked.</td>
</tr>
<tr>
<td>SET?</td>
<td>Returns a listing of instrument settings.</td>
</tr>
<tr>
<td>SOCKETServer:ENABLE</td>
<td>Enables or disables the socket server which supports a telnet or other TCPIP socket connection to send commands and queries to the instrument.</td>
</tr>
<tr>
<td>SOCKETServer:PORT</td>
<td>Sets the TCPIP port for the socket server connection.</td>
</tr>
<tr>
<td>SOCKETServer:PROTOCOL</td>
<td>Sets or queries the protocol for the socket server.</td>
</tr>
<tr>
<td>TEKSecure</td>
<td>Initializes both waveform and setup memories.</td>
</tr>
<tr>
<td>TIME?</td>
<td>Queries the time displayed by the instrument.</td>
</tr>
<tr>
<td>TIME:ZONE</td>
<td>Sets the time zone to the one specified.</td>
</tr>
<tr>
<td>TIME:ZONE:UTCDELTA</td>
<td>Sets or queries the time zone using the difference between the desired time zone and UTC.</td>
</tr>
<tr>
<td>TOTALuptime?</td>
<td>Returns the total number of hours the oscilloscope has been turned on since the NV memory was last programmed.</td>
</tr>
<tr>
<td>TOUCHSCREen:STATE</td>
<td>Sets or queries the enabled state of the touch screen.</td>
</tr>
<tr>
<td>*TRG</td>
<td>Performs the group execute trigger (GET).</td>
</tr>
<tr>
<td>*TST?</td>
<td>Tests the interface and returns status.</td>
</tr>
<tr>
<td>UNDO</td>
<td>Reverts the scope settings to a state before the previous command or user interface action.</td>
</tr>
<tr>
<td>UNLock</td>
<td>Unlocks front panel.</td>
</tr>
<tr>
<td>USBDevice:CONFigure</td>
<td>Used to configure the rear USB port to be off or enabled as a USBTMC device.</td>
</tr>
<tr>
<td>VERBose</td>
<td>Sets or queries the verbose state.</td>
</tr>
</tbody>
</table>
### Plot command group

Plot commands let you select the type and control the appearance of your plots.

#### Table 2-34: Plot commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLOT:ADDNew</td>
<td>Adds the specified plot.</td>
</tr>
<tr>
<td>PLOT:DELETE</td>
<td>Deletes the specified plot.</td>
</tr>
<tr>
<td>PLOT:LIST?</td>
<td>Lists all currently defined plots.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:BATHtub:BER</td>
<td>Sets or queries the bathtub BER value.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:BATHtub:XAXISUnits</td>
<td>Sets or queries the X-Axis unit, either unit intervals or seconds.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:BITType</td>
<td>Sets or queries the bit type to display for the specified eye diagram plot.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:LABEL:COLOR</td>
<td>Sets or queries the color of the specified trend label.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:LABEL:BOLD</td>
<td>Sets or queries the bold state of the specified trend label.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:LABEL:ITALIC</td>
<td>Sets or queries the italic state of the specified trend label.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:LABEL:SIZE</td>
<td>Sets or queries the font size of the specified trend label.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:LABEL:TYPE</td>
<td>Sets or queries the font type of the specified trend label.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:LABEL:UNDERLINE</td>
<td>Sets or queries the underline state of the specified trend label.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:LABEL.NAME</td>
<td>Sets or queries the specified trend's label.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:LABEL:XPOS</td>
<td>Sets or queries the x-position of the specified trend label.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:LABEL:YPOS</td>
<td>Sets or queries the y-position of the specified trend label.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:MASK?</td>
<td>Returns the name of the mask test associated with the specified eye diagram plot.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:NUMBins</td>
<td>Sets or queries the current histogram resolution.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:SOURCE&lt;x&gt;</td>
<td>Sets or queries the measurement source.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:SPECTRum:BASE</td>
<td>Sets or queries the spectrum base. Undefined for non-spectrum plots.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:SPECTRum:DYNRange</td>
<td>Sets or queries the dynamic range value.</td>
</tr>
<tr>
<td>PLOT:PLOT&lt;x&gt;:TYPE</td>
<td>Sets or queries the current plot type for the selected plot.</td>
</tr>
</tbody>
</table>

### Power command group

#### Table 2-35: Power commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWER:ADDNew</td>
<td>Adds the specified power measurement badge.</td>
</tr>
<tr>
<td>POWER:DELETE</td>
<td>Deletes the specified power measurement badge.</td>
</tr>
<tr>
<td>POWER:POWER&lt;x&gt;:AUTOSet</td>
<td>Executes power autoset for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWER:POWER&lt;x&gt;:CLRESPONSE:AMP[1-10]Val</td>
<td>Sets or queries the generator amplitude value of the specified configuration step for the Control Loop Response power measurement.</td>
</tr>
<tr>
<td>POWER:POWER&lt;x&gt;:CLRESPONSE:AMPMode</td>
<td>Sets or queries the amplitude mode for the Control Loop Response power measurement.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CLRESPONSE:CONSTAMPlitude</td>
<td>Sets or queries the constant amplitude voltage for the Control Loop Response power measurement.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CLRESPONSE:FREQ[1-11]Val</td>
<td>Sets or queries the generator frequency value of the specified configuration step for the Control Loop Response power measurement.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CLRESPONSE:GENerator</td>
<td>Sets or queries the generator source used to send stimulus signals to the DUT, for the Control Loop Response power measurement.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CLRESPONSE:IMPEDance</td>
<td>Sets or queries the vertical termination impedance for the Control Loop Response power measurement.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CLRESPONSE:INPUTSOurce</td>
<td>Sets or queries the input source for the Control Loop Response power measurement.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CLRESPONSE:OUTPUTSOurce</td>
<td>Sets or queries the output source for the Control Loop Response power measurement.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CLRESPONSE:PPD</td>
<td>Sets or queries the points per decade (PPD) value for the Control Loop Response power measurement.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CLRESPONSE:STARTFREQuency</td>
<td>Sets or queries the start frequency value for the Control Loop Response power measurement.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CLRESPONSE:STOPFREQuency</td>
<td>Sets or queries the stop frequency value for the Control Loop Response power measurement.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CYCLEAmp:INPUTSOurce</td>
<td>Sets or queries the input source for cycle amplitude measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CYCLEBase:INPUTSOurce</td>
<td>Sets or queries the input source for cycle base measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CYCLEMAX:INPUTSOurce</td>
<td>Sets or queries the input source for cycle maximum measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CYCLEMin:INPUTSOurce</td>
<td>Sets or queries the input source for cycle minimum measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CYCLEPKPK:INPUTSOurce</td>
<td>Sets or queries the input source for cycle peak-to-peak measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:CYCLETop:INPUTSOurce</td>
<td>Sets or queries the input source for cycle top measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:DIDT:INPUTSOurce</td>
<td>Sets or queries the input source for di/dt measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:DIDT:SOURCEEDGEType</td>
<td>Sets or queries the edge type for di/dt measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:DVDT:INPUTSOurce</td>
<td>Sets or queries the input source for dv/dt measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWeR:POWeRx&gt;:DVDT:SOURCEEDGEType</td>
<td>Sets or queries the edge type for dv/dt measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:EFFICIENCY:INPUTType</td>
<td>Sets or queries the input type (AC or DC) for efficiency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:EFFICIENCY:IOUT1SOURCE</td>
<td>Sets or queries the output 1 current source for efficiency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:EFFICIENCY:IOUT2SOURCE</td>
<td>Sets or queries the output 2 current source for efficiency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:EFFICIENCY:IOUT3SOURCE</td>
<td>Sets or queries the output 3 current source for efficiency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:EFFICIENCY:ISOUrce</td>
<td>Sets or queries the current source for efficiency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:EFFICIENCY:NUMOFOutputs</td>
<td>Sets or queries the number of outputs for efficiency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:EFFICIENCY:OUTPUT1Type</td>
<td>Sets or queries the Output1 type for efficiency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:EFFICIENCY:OUTPUT2Type</td>
<td>Sets or queries the Output2 type for efficiency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:EFFICIENCY:OUTPUT3Type</td>
<td>Sets or queries the Output3 type for efficiency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:EFFICIENCY:OUTPUTType</td>
<td>Sets or queries the Output type for efficiency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:EFFICIENCY:VOUT1SOURCE</td>
<td>Sets or queries the output 1 voltage source for efficiency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:EFFICIENCY:VOUT2SOURCE</td>
<td>Sets or queries the output 2 voltage source for efficiency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:EFFICIENCY:VOUT3SOURCE</td>
<td>Sets or queries the output 3 voltage source for efficiency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:EFFICIENCY:VSOURce</td>
<td>Sets or queries the voltage source for efficiency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:FREQUENCY:EDGE</td>
<td>Sets or queries the edge type for frequency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:FREQUENCY:INPUTSOurce</td>
<td>Sets or queries the input source for frequency measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:GATing</td>
<td>Sets or queries the gating type for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:GATing:GLOBAL</td>
<td>Sets or queries the gating settings for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:HARMONICS:CLASs</td>
<td>Sets or queries the class type for the harmonics measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:HARMONICS:CMETHod</td>
<td>Sets or queries the fundamental current method for the harmonics measurement of the specified power measurement number.</td>
</tr>
</tbody>
</table>
### Table 2-35: Power commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>POWer:POWer&lt;x&gt;:HARMONICS:FUNDCURRent</code></td>
<td>Sets or queries the fundamental current value for the harmonics measurement of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:HARMONICS:HORDer</code></td>
<td>Sets or queries the order value for the harmonics measurement of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:HARMONICS:HSOURce</code></td>
<td>Sets or queries the source type for the harmonics measurement of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:HARMONICS:iPOWer</code></td>
<td>Sets or queries the input power value for the harmonics measurement of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:HARMONICS:ISOURce</code></td>
<td>Sets or queries the current source for SOA measurement of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:HARMONICS:LINEFREQUENCY</code></td>
<td>Sets or queries the value for the line frequency for the power Harmonics measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:HARMONICS:ODDEVen</code></td>
<td>Sets or queries the harmonics value analysis format of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:HARMONICS:PFACtor</code></td>
<td>Sets or queries the value of power factor for the harmonics measurement of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:HARMONICS:POWERRating</code></td>
<td>Sets or queries the power level for the harmonics measurement of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:HARMONICS:RCURRent</code></td>
<td>Sets or queries the rated current for the harmonics measurement of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:HARMONICS:STANDard</code></td>
<td>Sets or queries the test mode for harmonics measurement of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:HARMONICS:STARTFREQUENCY</code></td>
<td>Sets or queries the value for the start frequency for the power Harmonics measurement. in the range of 1 Hz to 1 GHz.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:HARMONICS:UNITs</code></td>
<td>Sets or queries the harmonics results units of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:HARMONICS:VSOURce</code></td>
<td>Sets or queries the voltage source for SOA measurement of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:INDUCTANCE:EDGESource</code></td>
<td>Sets or queries the edge source for inductance measurement of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:INDUCTANCE:ISOURce</code></td>
<td>Sets or queries the current source for inductance measurement of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:INDUCTANCE:VSOURce</code></td>
<td>Sets or queries the voltage source for inductance measurement of the specified power measurement number.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:INPUTCAP:ISOURce</code></td>
<td>Sets or queries the inrush current input source of the specified Input Capacitance measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:INPUTCAP:PEAKCURRent</code></td>
<td>Sets or queries the peak current value of the specified Input Capacitance measurement.</td>
</tr>
</tbody>
</table>
### Table 2-35: Power commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWer:POWer&lt;x&gt;:INPUTCAP:PEAKvOLTage</td>
<td>Sets or queries the peak voltage value of the specified Input Capacitance measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:INPUTCAP:VSOURce</td>
<td>Sets or queries the input voltage source of the specified Input Capacitance measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:INRUSHcurrent:INPUTSOurce</td>
<td>Sets or queries the input source of the specified Inrush Current measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:INRUSHcurrent:PEAKCURRent</td>
<td>Sets or queries the peak current value of the specified Inrush Current measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:IVSINTEGRAL:ISOURce</td>
<td>Sets or queries the current source for I vs Integral V measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:IVSINTEGRAL:VSOURce</td>
<td>Sets or queries the voltage source for I vs Integral V measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:LABel</td>
<td>Sets or queries the custom name for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:LINERIPPLE:INPUTSOurce</td>
<td>Sets or queries the input source for line ripple measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:LINERIPPLE:LFREQuency</td>
<td>Sets or queries the frequency present for line ripple measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGNETICLOSS:ISOURce</td>
<td>Sets or queries the current source for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGNETICLOSS:VSOURce</td>
<td>Sets or queries the voltage source for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:AREAofcrosssection</td>
<td>Sets or queries the coil cross section area for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:EDGESOurce</td>
<td>Sets or queries the edge signal source for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:ISOURce</td>
<td>Sets or queries the current source for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:LENgth</td>
<td>Sets or queries the conductor length for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:PRIMARYTURNs</td>
<td>Sets or queries the number of primary turns for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:SEC1SOURce</td>
<td>Sets or queries the current source for secondary winding 1 for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:SEC1TURNs</td>
<td>Sets or queries the number of turns of secondary winding 1 for magnetic measurement in the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:SEC2SOURce</td>
<td>Sets or queries the current source for secondary winding 2 for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:SEC2TURNs</td>
<td>Sets or queries the number of turns of secondary winding 2 for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:SEC3SOURce</td>
<td>Sets or queries the current source for secondary winding 3 for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:SEC3TURNs</td>
<td>Sets or queries the number of turns of secondary winding 3 for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:SEC4SOURce</td>
<td>Sets or queries the current source for secondary winding 4 for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:SEC4TURNs</td>
<td>Sets or queries the number of turns of secondary winding 4 for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:SEC5SOURce</td>
<td>Sets or queries the current source for secondary winding 5 for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:SEC5TURNs</td>
<td>Sets or queries the number of turns of secondary winding 5 for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:SEC6SOURce</td>
<td>Sets or queries the current source for secondary winding 6 for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:SEC6TURNs</td>
<td>Sets or queries the number of turns of secondary winding 6 for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:SECWINDings</td>
<td>Sets or queries the number of secondary windings for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:UNITs</td>
<td>Sets or queries the units for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:MAGPROPERTY:VSOURce</td>
<td>Sets or queries the voltage source for magnetic measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:NDUTYCYCLE:EDGEType</td>
<td>Sets or queries the clock edge type for negative duty cycle measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:NDUTYCYCLE:INPUTSOurce</td>
<td>Sets or queries the input source for negative duty cycle measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:NPULSEWIDTH:INPUTSOurce</td>
<td>Sets or queries the input source for negative pulse width measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PDUTYCYCLE:EDGEType</td>
<td>Sets or queries the clock edge type for positive duty cycle measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PDUTYCYCLE:INPUTSOurce</td>
<td>Sets or queries the input source for positive duty cycle measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PERIOD:EDGE</td>
<td>Sets or queries the edge type for period measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PERIOD:INPUTSOurce</td>
<td>Sets or queries the input source for period measurement of the specified power measurement number.</td>
</tr>
</tbody>
</table>
### Table 2-35: Power commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWer:POWer&lt;x&gt;:POWERQUALITY:CCYCles</td>
<td>Sets or queries the calculate cycles over full cycles settings for the specified power quality measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:POWERQUALITY:FREference</td>
<td>Sets or queries the frequency reference type for power quality measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:POWERQUALITY:ISOURce</td>
<td>Sets or queries the current source for power quality measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:POWERQUALITY:VSOURce</td>
<td>Sets or queries the voltage source for power quality measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PRESET</td>
<td>Sets or queries the input source for positive pulse width measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PSRR:AMP[1-10]Val</td>
<td>Sets or queries the generator amplitude value of the specified configuration step for the Power Supply Rejection Ratio (PSRR) power measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PSRR:AMPMode</td>
<td>Sets or queries the amplitude mode for the Power Supply Rejection Ratio (PSRR) power measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PSRR:CONSTAMPlitude</td>
<td>Sets or queries the constant amplitude voltage for the Power Supply Rejection Ratio (PSRR) power measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PSRR:FREQ[1-11]Val</td>
<td>Sets or queries the generator frequency value of the specified configuration step for the Power Supply Rejection Ratio (PSRR) power measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PSRR:GENerator</td>
<td>Sets or queries the generator source used to send stimulus signals to the DUT for the Power Supply Rejection Ratio (PSRR) power measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PSRR:IMPEDance</td>
<td>Sets or queries the vertical termination impedance for the Power Supply Rejection Ratio (PSRR) power measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PSRR:INPUTSOurce</td>
<td>Sets or queries the input source for the Power Supply Rejection Ratio (PSRR) power measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PSRR:OUTPUTSOurce</td>
<td>Sets or queries the output source for the Power Supply Rejection Ratio (PSRR) power measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PSRR:PPD</td>
<td>Sets or queries the points per decade (PPD) value for the Power Supply Rejection Ratio (PSRR) power measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PSRR:STARTFREQuency</td>
<td>Sets or queries the start frequency value for the Power Supply Rejection Ratio (PSRR) power measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:PSRR:STOPFREQuency</td>
<td>Sets or queries the stop frequency value for the Power Supply Rejection Ratio (PSRR) power measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:RDSON:DEVICEType</td>
<td>Sets or queries the device type for the power drain source on resistance measurement for RDSon measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:RDSON:ISOURce</td>
<td>Sets or queries the current source for RDSon measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:RDSON:VSOURce</td>
<td>Sets or queries the voltage source for RDSon measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: ABSolute:FALLHigh</td>
<td>Sets or queries the falling edge for high reference level in absolute units for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: ABSolute:FALLLow</td>
<td>Sets or queries the falling edge for low reference level in absolute units for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: ABSolute:FALLMid</td>
<td>Sets or queries the falling edge for mid reference level in absolute units for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: ABSolute:HYSTeresis</td>
<td>Sets or queries the absolute hysteresis value for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: ABSolute:RISEHigh</td>
<td>Sets or queries the rising edge for high reference level in absolute units for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: ABSolute:RISELow</td>
<td>Sets or queries the rising edge for low reference level in absolute units for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: ABSolute:RISEMid</td>
<td>Sets or queries the rising edge for mid reference level in absolute units for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: BASETop</td>
<td>Sets or queries the reference level base top method for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: METHod</td>
<td>Sets or queries the method to configure reference level values for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: PERCent:FALLHigh</td>
<td>Sets or queries the falling edge for high reference level in percentage for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: PERCent:FALLLow</td>
<td>Sets or queries the falling edge for low reference level in percentage for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: PERCent:FALLMid</td>
<td>Sets or queries the falling edge for mid reference level in percentage for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: PERCent:HYSTeresis</td>
<td>Sets or queries the hysteresis in percentage for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: PERCent:RISEHigh</td>
<td>Sets or queries the rising edge for high reference level in percentage for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: PERCent:RISELow</td>
<td>Sets or queries the rising edge for low reference level in percentage for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: PERCent:RISEMid</td>
<td>Sets or queries the rising edge for mid reference level in percentage for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:REFLevels: PERCent:TYPE</td>
<td>Sets or queries the reference levels for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:RESULTS: ALLAcqs:MAXimum?</td>
<td>Queries the maximum value of all acquisitions for the measurement parameter of the specified power measurement number.</td>
</tr>
</tbody>
</table>
### Table 2-35: Power commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:ALLAcqs:MEAN?</td>
<td>Queries the mean value of all acquisitions for the measurement parameter of the specified power measurement number.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:ALLAcqs:MINimum?</td>
<td>Queries the minimum value of all acquisitions for the measurement parameter of the specified power measurement number.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:ALLAcqs:PK2PK?</td>
<td>Queries the peak-to-peak value of all acquisitions for the measurement parameter of the specified power measurement number.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:ALLAcqs:POPUlation?</td>
<td>Queries the population (number of complete cycles) of all acquisitions for the measurement parameter of the specified power measurement number.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:ALLAcqs:STDDev?</td>
<td>Queries the standard deviation value of all acquisitions for the measurement parameter of the specified power measurement number.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:CURRentacq:F1MAG?</td>
<td>Queries the first harmonics magnitude value for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:CURRentacq:F3MAG?</td>
<td>Queries the third harmonics magnitude value for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:CURRentacq:FREQUENCY?</td>
<td>Queries the fundamental frequency for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:CURRentacq:IRMS?</td>
<td>Queries the RMS current value for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:CURRentacq:MAXimum?</td>
<td>Queries the maximum value of the current acquisition for the measurement parameter of the specified power measurement number.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:CURRentacq:MEAN?</td>
<td>Queries the mean value of the current acquisition for the measurement parameter of the specified power measurement number.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:CURRentacq:MINimum?</td>
<td>Queries the minimum value of the current acquisition for the measurement parameter of the specified power measurement number.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:CURRentacq:PK2PK?</td>
<td>Queries the peak-to-peak value of the current acquisition for the measurement parameter of the specified power measurement number.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:CURRentacq:POHCL?</td>
<td>Queries the limit of partial odd harmonic current for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:CURRentacq:POHCM?</td>
<td>Queries the measured value of partial odd harmonic current for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:CURRentacq:POHCS?</td>
<td>Queries the status of partial odd harmonic current for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:CURRentacq:POPUlation?</td>
<td>Queries the population (number of complete cycles) of the current acquisition for the measurement parameter of the specified power measurement number.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:CURRentacq:RMS?</td>
<td>Queries the RMS value of the source selected for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWER:POWer&lt;x&gt;:RESULTS:CURRentacq:STATUS?</td>
<td>Queries the status of the measurement for the specified power measurement badge.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:RESULTS:CURRentacq:STDDev?</td>
<td>Queries the standard deviation value of the current acquisition for the measurement parameter of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:RESULTS:CURRentacq:THDF?</td>
<td>Queries the total harmonic distortion (fundamental) value for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:RESULTS:CURRentacq:THDR?</td>
<td>Queries the total harmonic distortion (RMS) value for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:RESULTS:CURRentacq:TRPWR?</td>
<td>Queries the true power value for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:RESULTS:CURRentacq:VRMS?</td>
<td>Queries the RMS voltage value for the specified power measurement badge.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SEQSETup</td>
<td>Runs the power measurement sequence setup.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SEQuence</td>
<td>Sets or queries the run state of a single sequence power measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SOA:ISOURce</td>
<td>Sets or queries the current source for SOA measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SOA:POINT</td>
<td>Sets or queries the X or Y coordinate value for a specified SOA mask point.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SOA:RECAIlmask</td>
<td>Recalls or queries the recall mask file name of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SOA:RECAIlmask:FILENAME</td>
<td>Sets or queries the file name for saving SOA mask file name of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SOA:SAVemask:AUTOINCrement</td>
<td>Sets or queries the state of auto-increment for saved SOA mask file names of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SOA:SAVemask:FILENAME</td>
<td>Sets or queries the mask file name for SOA measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SOA:SAVemask:FOLDER</td>
<td>Sets or queries the mask file folder path for SOA measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SOA:VSOURce</td>
<td>Sets or queries the voltage source for SOA measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGLOSS:DEVICEType</td>
<td>Sets or queries the conduction calculation method for switching loss measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGLOSS:GATESOurce</td>
<td>Sets or queries the gate voltage (V_g) for the switching loss measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGLOSS:ILEVELAbs</td>
<td>Sets or queries the current level (Ton-Start &amp; Stop) in absolute units for switching loss measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGLOSS:ILEVELPct</td>
<td>Sets or queries the current level (Ton-Start &amp; Stop) in percentage for switching loss measurement of the specified power measurement number.</td>
</tr>
</tbody>
</table>
Table 2-35: Power commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGLOSS:ISOURce</td>
<td>Sets or queries the current source for the switching loss measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGLOSS:LEVELUNIts</td>
<td>Sets or queries the level units for switching loss measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGLOSS:RDSOn</td>
<td>Sets or queries the RDS(on) value for switching loss measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGLOSS:SWLCONFIGType</td>
<td>Sets or queries the configuration type for the switching loss measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGLOSS:VCESat</td>
<td>Sets or queries the value for the VCE(sat) value for switching loss measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGLOSS:VGLevel</td>
<td>Sets or queries the gate voltage value (Vg Level Ton-Start) for the switching loss measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGLOSS:VLEVELAbs</td>
<td>Sets or queries the voltage level (Ton-Start &amp; Stop) in absolute units for switching loss measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGLOSS:VLEVELPct</td>
<td>Sets or queries the voltage level (Ton-Start &amp; Stop) in percentage for switching loss measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGLOSS:VSOURce</td>
<td>Sets or queries the voltage source for the switching loss measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGRIPPLE:INPUTSOurce</td>
<td>Sets or queries the input source for switching ripple measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:SWITCHINGRIPPLE:LFREQuency</td>
<td>Sets or queries the switching frequency for switching ripple measurement of the specified power measurement number.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:TURNOFFtime:FREQuency</td>
<td>Sets or queries the input frequency used by the AC or DC converter of the specified Turn Off Time measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:TURNOFFtime:INPUTLEVel</td>
<td>Sets or returns the input frequency used by the AC or DC converter of the specified Turn Off Time measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:TURNOFFtime:NUMOUTputs</td>
<td>Sets or queries the number of outputs of the specified Turn Off Time power measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:TURNOFFtime:OUTPUT1SOURce</td>
<td>Sets or queries the output 1 source of the specified Turn Off Time measurement.</td>
</tr>
<tr>
<td>POWer:POWer&lt;x&gt;:TURNOFFtime:OUTPUT1VOLTage</td>
<td>Sets or queries the output 1 voltage level of the of the specified Turn Off Time power measurement.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOFFtime:OUTPUT2SOURce</code></td>
<td>Sets or queries the output 2 source of the specified Turn Off Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOFFtime:OUTPUT2VOLTage</code></td>
<td>Sets or queries the output 2 voltage level of the specified Turn Off Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOFFtime:OUTPUT3SOURce</code></td>
<td>Sets or queries the output 3 source of the specified Turn Off Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOFFtime:OUTPUT3VOLTage</code></td>
<td>Sets or queries the output 3 voltage level of the specified Turn Off Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOFFtime:OUTPUT4SOURce</code></td>
<td>Sets or queries the output 4 source of the specified Turn Off Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOFFtime:OUTPUT4VOLTage</code></td>
<td>Sets or queries the output 4 voltage level of the specified Turn Off Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOFFtime:OUTPUT5SOURce</code></td>
<td>Sets or queries the output 5 source of the specified Turn Off Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOFFtime:OUTPUT5VOLTage</code></td>
<td>Sets or queries the output 5 voltage level of the specified Turn Off Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOFFtime:OUTPUT6SOURce</code></td>
<td>Sets or queries the output 6 source of the specified Turn Off Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOFFtime:OUTPUT6VOLTage</code></td>
<td>Sets or queries the output 6 voltage level of the specified Turn Off Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOFFtime:OUTPUT7SOURce</code></td>
<td>Sets or queries the output 7 source of the specified Turn Off Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOFFtime:OUTPUT7VOLTage</code></td>
<td>Sets or queries the output 7 voltage level of the specified Turn Off Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOFFtime:TYPE</code></td>
<td>Sets or queries the type of AC/DC converter used in the specified Turn Off Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOnTime:FREQuency</code></td>
<td>Sets or queries the input frequency used by the AC or DC converter of the specified Turn On Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOnTime:INPUTLEVEL</code></td>
<td>Sets or queries the input voltage level of the specified Turn On Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOnTime:INPUTSOURce</code></td>
<td>Sets or queries the input source of the specified Turn On Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNOnTime:MAXTime</code></td>
<td>Sets or queries the maximum turn on time of the specified Turn On Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:MAXVoltage</code></td>
<td>Sets or queries the maximum voltage setting of the specified Turn On Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:NUMOUTPUTs</code></td>
<td>Sets or queries the number of outputs for the specified Turn On Time power measurement.</td>
</tr>
</tbody>
</table>
## Command groups

### Table 2-35: Power commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:OUTPUT1SOURce</code></td>
<td>Sets or queries the output 1 source of the specified Turn On Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:OUTPUT1VOLTage</code></td>
<td>Sets or queries the output 1 voltage level of the specified Turn On Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:OUTPUT2SOURce</code></td>
<td>Sets or queries the output 2 source of the specified Turn On Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:OUTPUT2VOLTage</code></td>
<td>Sets or queries the output 2 voltage level of the specified Turn On Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:OUTPUT3SOURce</code></td>
<td>Sets or queries the output 3 source of the specified Turn On Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:OUTPUT3VOLTage</code></td>
<td>Sets or queries the output 3 voltage level of the specified Turn On Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:OUTPUT4SOURce</code></td>
<td>Sets or queries the output 4 source of the specified Turn On Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:OUTPUT4VOLTage</code></td>
<td>Sets or queries the output 4 voltage level of the specified Turn On Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:OUTPUT5SOURce</code></td>
<td>Sets or queries the output 5 source of the specified Turn On Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:OUTPUT5VOLTage</code></td>
<td>Sets or queries the output 5 voltage level of the specified Turn On Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:OUTPUT6SOURce</code></td>
<td>Sets or queries the output 6 source of the specified Turn On Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:OUTPUT6VOLTage</code></td>
<td>Sets or queries the output 6 voltage level of the specified Turn On Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:OUTPUT7SOURce</code></td>
<td>Sets or queries the output 7 source of the specified Turn On Time measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:OUTPUT7VOLTage</code></td>
<td>Sets or queries the output 7 voltage level of the specified Turn On Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TURNONtime:TYPE</code></td>
<td>Sets or queries the type of AC/DC converter used in the specified Turn On Time power measurement.</td>
</tr>
<tr>
<td><code>POWer:POWer&lt;x&gt;:TyPe</code></td>
<td>Sets or queries the measurement type for the specified power measurement badge.</td>
</tr>
<tr>
<td><code>POWERTABle:ADDNew</code></td>
<td>Adds the power harmonics table.</td>
</tr>
<tr>
<td><code>POWERTABle:DELete</code></td>
<td>Deletes the power harmonics table.</td>
</tr>
<tr>
<td><code>POWERTABle:LISfT?</code></td>
<td>Lists all defined power harmonics tables.</td>
</tr>
</tbody>
</table>
Save and Recall command group

Use the commands in the Save and Recall Command Group to store and retrieve internal waveforms and settings. When you save a setup, you save all the settings of the instrument. When you recall a setup, the instrument restores itself to the state that it was in when you originally saved that setting.

Table 2-36: Save and Recall commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FACTory</td>
<td>Resets the instrument to factory default settings.</td>
</tr>
<tr>
<td>RECALL:SESSION</td>
<td>Restores the state of the instrument from a saved session file.</td>
</tr>
<tr>
<td>RECALL:SETUP</td>
<td>Recalls saved instrument settings.</td>
</tr>
<tr>
<td>RECALL:WAVEFORM</td>
<td>Recalls a stored waveform to a reference memory location.</td>
</tr>
<tr>
<td>SAVE:EVENTtable:BUS</td>
<td>Saves bus results table to the specified file.</td>
</tr>
<tr>
<td>SAVE:EVENTtable:CUSTOM</td>
<td>Saves the results table to the specified file path and name.</td>
</tr>
<tr>
<td>SAVE:EVENTtable:CUSTOM:COMMENTS</td>
<td>Sets or queries comments to be included in saved results table files.</td>
</tr>
<tr>
<td>SAVE:EVENTtable:CUSTOM:DATAFORMAT</td>
<td>Sets or queries the data format to use for saving results table data.</td>
</tr>
<tr>
<td>SAVE:EVENTtable:CUSTOM:INCLUDEREFs</td>
<td>Sets or queries whether to include displayed reference waveforms with saved results table files.</td>
</tr>
<tr>
<td>SAVE:EVENTtable:MEASUREMENT</td>
<td>Saves data (measurement) results to the specified file.</td>
</tr>
<tr>
<td>SAVE:IMAGE</td>
<td>Saves a capture of the screen contents to the specified image file.</td>
</tr>
<tr>
<td>SAVE:REPORT</td>
<td>Saves a report to the specified file or, if no argument is specified, uses the folder and file name specified by the related commands.</td>
</tr>
<tr>
<td>SAVE:REPORT:COMMENTS</td>
<td>Sets or queries the comments to be included in saved report files.</td>
</tr>
<tr>
<td>SAVE:SESSION</td>
<td>Saves the state of the instrument, including reference waveforms, to a saved session file.</td>
</tr>
<tr>
<td>SAVE:SETUP</td>
<td>Saves the current instrument state to the specified file.</td>
</tr>
<tr>
<td>SAVE:SETUP:INCLUDEREFs</td>
<td>Sets or queries whether displayed reference waveforms are to be included in saved setups.</td>
</tr>
<tr>
<td>SAVE:WAVEFORM</td>
<td>Saves the specified waveform(s) to the specified destination file(s).</td>
</tr>
<tr>
<td>SAVE:WAVEFORM:SOURCELIST?</td>
<td>Returns a list of the available waveforms that can be specified as the source for the SAVE:WAVEFORM command.</td>
</tr>
</tbody>
</table>
Save On command Group

Use this group of commands to program the oscilloscope to save images, measurements, waveforms, or the instrument setup, on triggers that you select.

Table 2-37: Save On commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVEON:FILE:DEST</td>
<td>Sets or queries the location where files are saved.</td>
</tr>
<tr>
<td>SAVEON:FILE:NAME</td>
<td>Sets or queries the file name to use when SAVEON:TRIGger is ON.</td>
</tr>
<tr>
<td>SAVEON:IMAGE:FILEFormat</td>
<td>Sets or queries the file format to be used for saved image files.</td>
</tr>
<tr>
<td>SAVEON:IMAGE</td>
<td>Sets or queries whether to save a screen capture when a trigger occurs.</td>
</tr>
<tr>
<td>SAVEON:TRIGger</td>
<td>Sets or queries whether to save a file when a trigger occurs.</td>
</tr>
<tr>
<td>SAVEON:WAVEform</td>
<td>Sets or queries whether to save a waveform when a limit test failure, mask failure, or trigger occurs.</td>
</tr>
<tr>
<td>SAVEON:WAVEform:FILEFormat</td>
<td>Sets or queries the file format for saving waveform.</td>
</tr>
<tr>
<td>SAVEON:WAVEform:SOURce</td>
<td>Sets or queries the sources for saving waveforms.</td>
</tr>
</tbody>
</table>
**Search and Mark command group**

Use search and mark commands to seek out and identify information in waveform records that warrant further investigation.

Table 2-38: Search and Mark commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:ADDNew</td>
<td>Adds the specified search.</td>
</tr>
<tr>
<td>SEARCH:DELETE</td>
<td>Deletes the specified search.</td>
</tr>
<tr>
<td>SEARCH:LST?</td>
<td>Lists all currently defined searches.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:COPY</td>
<td>Copies the search criteria to or from the trigger.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:NAVigate</td>
<td>Sets the navigation action for search marks.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TOTAL?</td>
<td>Queries the total number of found search marks for this search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ARINC429A:CONDition</td>
<td>Specifies a field or condition for an ARINC429 bus to search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ARINC429A:ERRTYPe</td>
<td>Sets or queries the error type when searching on an ARINC429 bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ARINC429A:LABel:QUALifier</td>
<td>Sets or queries the qualifier to be used when searching on label data for an ARINC429 bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ARINC429A:LABel:VALue</td>
<td>Sets or queries the low value when searching on an ARINC429 label field.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ARINC429A:LABel:HIVALue</td>
<td>Sets or queries the high value when searching on an ARINC429 label field.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ARINC429A:SDI:VALue</td>
<td>Sets or queries the the when searching on an ARINC429 SDI field.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ARINC429A:SSM:VALue</td>
<td>Sets or queries the the when searching on an ARINC429 SSM field.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ARINC429A:DATa:QUALifier</td>
<td>Sets or queries the qualifier to be used when searching on data in the DATA field for an ARINC429 bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ARINC429A:DATa:VALue</td>
<td>Sets or queries the low value when searching on an ARINC429 data field.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ARINC429A:DATa:HIVALue</td>
<td>Sets or queries the high value when searching on an ARINC429 data field.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:AUDio:CONDition</td>
<td>Sets or queries the condition (word select / frame sync, or matching data) to be used when searching on an audio bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:AUDio:DATa:HITDMVALue</td>
<td>Sets or queries the binary data string for the high data word to be used when searching on a TDM audio bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:AUDio:DATa:HIVALue</td>
<td>Sets or queries the binary data string for the high data word to be used when searching on an audio bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:AUDio:DATa:OFFSet</td>
<td>Sets or queries the data offset value (TDM channel) to be used when searching on a TDM type audio bus signal.</td>
</tr>
</tbody>
</table>
### Table 2-38: Search and Mark commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:AUDIO:DATa:QUALifier</td>
<td>Sets or queries the qualifier to be used when searching on an audio bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:AUDIO:DATa:TDMVALue</td>
<td>Sets or queries the binary data string for the single or low data word to be used when searching on a TDM audio bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:AUDIO:DATa:VALue</td>
<td>Sets or queries the binary data string for the single or low data word to be used when searching on an audio bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:AUDIO:DATa:WORD</td>
<td>Sets or queries the alignment of the data (left, right or either) to be used when searching on a non-TDM type audio bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:CAN:CONDition</td>
<td>Sets or queries the CAN bus trigger condition.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:CAN:DATa:OFFSet</td>
<td>Sets or queries the data offset value, in bytes, to use when searching on the CAN data field.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:CAN:DATa:SIZe</td>
<td>Sets or queries the CAN bus trigger data size.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:CAN:DATa:VALue</td>
<td>Sets or queries the binary data value to be used when searching on a CAN bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:CAN:ERRType</td>
<td>Sets or queries the type of error condition for a CAN bus to search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:CAN:FD:BRSBit</td>
<td>Sets or queries the value of the bit rate switch bit (BRS bit) for a CAN bus to search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:CAN:FD:ESIBit</td>
<td>Sets or queries the value of the error state indicator bit (ESI bit) for a CAN bus to search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:CAN:FRAMEtype</td>
<td>Sets or queries CAN bus trigger frame type.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:CAN:IDenti:MODE</td>
<td>Sets or queries the CAN bus trigger identifier mode.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ETHERnet:CONDition</td>
<td>Specifies a field or condition within an Ethernet frame to search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ETHERnet:DATa:HIVALue</td>
<td>Sets or queries the binary data value to be used when searching on an Ethernet bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ETHERnet:DATa:OFFSet</td>
<td>Specifies the data offset value, in bytes, to use when searching on the Ethernet data field.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS: ETHERnet:DATa:QUALifier</td>
<td>Sets the qualifier to be used when searching on an Ethernet bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS: ETHERnet:DATa:VALUE</td>
<td>Specifies the binary value to use when searching on the Ethernet data field.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS: FLEXRAY:CYCLEcount:HIVALue</td>
<td>Sets or queries the FlexRay bus search cycle count value.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS: FLEXRAY:EOFTYPE</td>
<td>Sets or queries the FlexRay bus search trigger end of file type.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS: FLEXRAY:ERRTYPE</td>
<td>Sets or queries the FlexRay bus search trigger error type.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS: FLEXRAY:FRAMEID:HIVALue</td>
<td>Sets or queries the high value when searching on a FlexRay bus frame id field.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS: FLEXRAY:FRAMEID:QUALifier</td>
<td>Sets the qualifier to be used when searching on a FlexRay bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS: FLEXRAY:FRAMEType</td>
<td>Sets or queries the FlexRay bus search trigger frame type.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS: FLEXRAY:HEADER:CRC</td>
<td>Sets or queries the FlexRay bus search trigger header CRC.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:I2C: ADDRess:MODE</td>
<td>Sets or queries the address mode for the specified I2C bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:I2C: ADDRess:VA:LUe</td>
<td>Sets or queries the address string when the search condition for the specified search is Address or AddressData, to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:I2C: CONDITION</td>
<td>Sets or queries the trigger condition for the specified I2C bus trigger search to determine where to place a mark.</td>
</tr>
</tbody>
</table>
Table 2-38: Search and Mark commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:I2C:DATa:DIREction</td>
<td>Sets or queries the direction of the data for the I2C bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:I2C:DATa:SIZE</td>
<td>Sets or queries the length of the data string in bytes used for the specified I2C bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:I2C:DATa:VALUE</td>
<td>Sets or queries the data value of the data token for the specified I2C bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:LIN:DATa:HIVALue</td>
<td>Specifies the high data value to be used in a LIN search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:LIN:DATa:SIZe</td>
<td>Sets or queries the LIN bus search trigger data size.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:LIN:DATa:VALUE</td>
<td>Sets or queries the LIN bus search trigger data value.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:LIN:ERRTYPE</td>
<td>Sets or queries the LIN bus search trigger error type.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:MIL1553B:CONDition</td>
<td>Sets or queries the field or condition for a MIL-STD-1553 bus to search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:MIL1553B:ERRTYPE</td>
<td>Sets or queries the type of error condition for a MIL-STD-1553 bus to search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:MIL1553B:COMMAND:TRBit</td>
<td>Sets or queries the value of the command word Transmit / Receive bit for a MIL-STD-1553 bus to search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:MIL1553B:COMMAND:PARity</td>
<td>Sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:MIL1553B:COMMAND:COUNT</td>
<td>Sets or queries the value of the command word &quot;word count&quot; field for a MIL-STD-1553 bus to search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:MIL1553B:COMMAND:SUBAddress</td>
<td>Sets or queries the value of the command word subaddress field for a MIL-STD-1553 bus to search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:QUALifier</td>
<td>Sets or queries the qualifier to be used when searching on command word addresses for a MIL-STD-1553 bus.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:HIVALue</td>
<td>Sets or queries the high value when searching on command word addresses for a MIL-STD-1553 bus.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:MIL1553B:DATa:PARity</td>
<td>Sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to search on.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGGER:A:BUS:MIL1553B:STATUS:PARITY</td>
<td>Sets or queries the value of the status word parity bit for a MIL-STD-1553 bus to search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGGER:A:BUS:MIL1553B:STATUS:ADDRESS:QUALIFIER</td>
<td>Sets or queries the qualifier to be used when searching on status word addresses for a MIL-STD-1553 bus.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGGER:A:BUS:MIL1553B:STATUS:ADDRESS:HIVALUE</td>
<td>Sets or queries the high value when searching on status word addresses for a MIL-STD-1553 bus.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:BCR</td>
<td>Sets or queries the value of the broadcast command received bit (BCR bit, bit 15) in a status word for a MIL-STD-1553 bus to search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:DBCA</td>
<td>Sets or queries the value of the dynamic bus control acceptance bit (DBCA bit, bit 18) in a status word for a MIL-STD-1553 bus to search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGGER:A:BUS:PARALLEL:DATA:VALUE</td>
<td>Sets or queries the data value for a parallel bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGGER:A:BUS:RS232C:DATA:VALUE</td>
<td>Sets or queries the data string value for the specified RS232c bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGGER:A:BUS:RS232C:CONDITION</td>
<td>Sets or queries the condition for an RS232C bus search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGGER:A:BUS:RS232C:SIZE</td>
<td>Sets or queries the length of the data string in bytes to be used for an RS232 bus search to determine where to place a mark when the search condition is Data.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGGER:A:BUS:SENT:CONDITION</td>
<td>Sets or queries the search condition for a SENT bus.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGGER:A:BUS:SENT:ERRORTYPE</td>
<td>Sets or queries the error type to be used when searching on SENT data.</td>
</tr>
<tr>
<td>Command: SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: ERRType:_CRC</td>
<td>Sets or queries the CRC error type to be used when searching on SENT data.</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: FAST:CHAN1A:HIVALue</td>
<td>Sets or queries the high binary fast channel 1 value to use when searching on a SENT bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: FAST:CHAN1A:QUALifier</td>
<td>Sets or queries the qualifier to be used when searching on SENT fast packet bus data for device channel 1.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: FAST:CHAN1A:VALue</td>
<td>Sets or queries the binary fast channel 1 value to be used when searching on a SENT bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: FAST:CHAN2B:HIVALue</td>
<td>Sets or queries the high binary fast channel 2 value to use when searching on a SENT bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: FAST:CHAN2B:QUALifier</td>
<td>Sets or queries the qualifier to be used when searching on SENT fast packet bus data for device channel 2.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: FAST:CHAN2B:VALue</td>
<td>Sets or queries the binary fast channel 2 value to be used when searching on a SENT bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: FAST:COUNTer:HIVALue</td>
<td>Sets or queries the high binary fast message counter value to use when searching on a SENT bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: FAST:COUNTer:QUALifier</td>
<td>Sets or queries the qualifier to be used when searching on SENT fast packet bus data for the secure format counter.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: FAST:COUNTer:VALue</td>
<td>Sets or queries the binary fast message counter value to be used when searching on a SENT bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: FAST:INVERTNIBble:VALue</td>
<td>Sets or queries the binary fast message inverted nibble value to be used when searching on a SENT bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: FAST:STATus:VALue</td>
<td>Sets or queries the binary status value to be used when searching on a SENT bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: PAUSE:QUALifier</td>
<td>Sets or queries the qualifier to be used when searching on SENT pause pulses.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: PAUSE:TICKs:HIVALue</td>
<td>Sets or queries the maximum number of pause clock ticks to be used when searching on a SENT bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: PAUSE:TICKs:VALue</td>
<td>Sets or queries the minimum number of pause clock ticks to be used when searching on a SENT bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: SLOW:DATA:HIVALue</td>
<td>Sets or queries the high binary Slow channel data value to use when searching on SENT Slow packet bus data.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: SLOW:DATA:QUALifier</td>
<td>Sets or queries the qualifier to be used when searching on SENT Slow packet bus data.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: SLOW:DATA:VALue</td>
<td>Sets or queries the binary slow channel data value to be used when searching on SENT Slow packet bus data.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SENT: SLOW:IDenti fier:VALue</td>
<td>Sets or queries the binary Slow identifier value to be used when searching on SENT Slow packet bus data.</td>
</tr>
</tbody>
</table>
### Table 2-38: Search and Mark commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SOUrce</td>
<td>Sets or queries the bus source for the specified bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPI:CONDition</td>
<td>Sets or queries the trigger condition for the specified SPI bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPI:DATa:SIZE</td>
<td>Sets or queries the length of the data string in bytes used for the specified SPI bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPI:DATa:VALue</td>
<td>Sets or queries the data value of the data token for the specified SPI bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPMI:CONDition</td>
<td>This command sets or queries the search condition for an SPMI bus.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPMI:DATa:VALue</td>
<td>This command specifies the binary data string used for SPMI when the search condition is MASTERREAD, MASTERWRITE, REGREAD, REGWRITE, EXTREGREAD, EXTREGWRITE, LONGEXTREGREAD, LONGEXTREGWRITE, or REG0WRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPMI:MASTERADDRess:VALue</td>
<td>This command sets or queries the binary data string that identifies the master address used in SPMI search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPMI:NORESPonse</td>
<td>This command sets or queries whether or not to search for No Response frames.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPMI:REGISTERADDRess:VALue</td>
<td>This command sets or queries the binary data string that identifies the register address used in SPMI triggering.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPMI:SLAVEADDRess:VALue</td>
<td>This command sets or queries the binary data string that identifies the slave address used in SPMI.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:USB:ADDress:HIVALue</td>
<td>Sets or queries the address value for normal token to be used with In Range and Out of Range qualifiers for the specified USB bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:USB:ADDress:VALue</td>
<td>Sets or queries the address value for normal token for the specified USB bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:USB:CONDition</td>
<td>Sets or queries the search condition for the specified USB bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:USB:DATa:HIVALue</td>
<td>Sets or queries the data value for data token used with In Range and Out of Range qualifiers for the specified USB bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:USB:DATa:OFFSet</td>
<td>Sets or queries data offset for the specified USB bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:USB:DATa:SIZe</td>
<td>Sets or queries the length of the data string in bytes for the specified USB bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:USB:DATa:TYPe</td>
<td>Sets or queries the data packet type for the specified USB bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:USB:DATa:VALue</td>
<td>Sets or queries the data value for data token for the specified bus trigger search to determine where to place a mark.</td>
</tr>
</tbody>
</table>
### Table 2-38: Search and Mark commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ENDPoint:VALUE</td>
<td>Sets or queries the endpoint value for normal token for the specified USB bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:ERRTYPE</td>
<td>Sets or queries the error type for the specified USB bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:HANDSHAKEType</td>
<td>Sets or queries the handshake type for the specified USB bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SOFTRAMENUMber</td>
<td>Sets or queries the frame number string to use for the Start of Frame for the specified USB bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPECIALType</td>
<td>Sets or queries the PID value for the USB bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPLIT:ET:VALUE</td>
<td>Sets or queries the Endpoint Type value for the specified USB bus trigger split token field search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPLIT:HUB:VALUE</td>
<td>Sets or queries the hub address of the specified USB bus trigger on split token field search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPLIT:PORT:VALUE</td>
<td>Sets or queries the port address for the specified USB bus trigger on split token field search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPLIT:SC:VALUE</td>
<td>Sets or queries the Start/Complete value for the specified USB bus trigger on split token field search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPLIT:SE:VALUE</td>
<td>Sets or queries the Start/End value for the specified USB bus trigger on split token field search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:TOKENType</td>
<td>Sets or queries the token type when the specified USB bus trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:EDGE:SLOpe</td>
<td>Sets or queries the slope for an edge trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREAD:DATARate</td>
<td>Sets or queries the DDR read search data rate for DDR3 and LPDDR3 standards of the specified search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREAD:DATAsource</td>
<td>Sets or queries the DDR read data source when the search type is DDR READ.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREAD:HYSteresis</td>
<td>Sets or queries the DDR read hysteresis reference level value, when the search type is DDR READ.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREAD:MARgin</td>
<td>Sets or queries the DDR read margin reference level value, when the search type is DDR READ.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREAD:POSTAMBLE:LENGTH</td>
<td>Sets or queries the DDR read postamble length when the search type is DDR READ.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREAD:PREAMBLE:TYPE</td>
<td>Sets or queries the DDR read preamble type when the search type is DDR READ.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH</td>
<td>Sets or queries the DDR read data high reference level value, when the search type is DDR READ.</td>
</tr>
</tbody>
</table>
### Table 2-38: Search and Mark commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREAD:REFLevel:DATA:LOW</td>
<td>Sets or queries the DDR read data low reference level value, when the search type is DDR READ.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREAD:REFLevel:DATA:MID</td>
<td>Sets or queries the DDR read data mid reference level value, when the search type is DDR READ.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH</td>
<td>Sets or queries the DDR read reference level strobe high value, when the search type is DDR READ.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW</td>
<td>Sets or queries the DDR read reference level strobe low value, when the search type is DDR READ.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREAD:REFLevel:STROBE:MID</td>
<td>Sets or queries the DDR read reference level strobe mid value, when the search type is DDR READ.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREAD:STANdard</td>
<td>Sets or queries the DDR read search standard as DDR3 or LPDDR3.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREAD:STROBESource</td>
<td>Sets or queries the DDR read strobe source when the search type is DDR READ.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:DATARate</td>
<td>Sets or queries the DDR read/write data rate for DDR3 and LPDDR3 standards of the specified search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:DATASource</td>
<td>Sets or queries the DDR read/write data source when the search type is DDR READWRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:HYSteresis</td>
<td>Sets or queries the DDR read/write hysteresis reference level value, when the search type is DDR READWRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:MARgin</td>
<td>Sets or queries the DDR read/write margin reference level value, when the search type is DDR READWRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth</td>
<td>Sets or queries the DDR read/write postamble length when the search type is DDR READWRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE</td>
<td>Sets or queries the DDR read/write preamble type when the search type is DDR READWRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH</td>
<td>Sets or queries the DDR read/write data high reference level value, when the search type is DDR READWRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW</td>
<td>Sets or queries the DDR read/write data low reference level value, when the search type is DDR READWRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID</td>
<td>Sets or queries the DDR read/write data mid reference level value, when the search type is DDR READWRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH</td>
<td>Sets or queries the DDR read/write reference level strobe high value, when the search type is DDR READWRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW</td>
<td>Sets or queries the DDR read/write reference level strobe low value, when the search type is DDR READWRITE.</td>
</tr>
</tbody>
</table>
Table 2-38: Search and Mark commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID</td>
<td>Sets or queries the DDR read/write reference level strobe mid value, when the search type is DDR READWRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:REFLEVELMode</td>
<td>Sets or queries the DDR read/write reference level mode to auto or manual, when the search type is DDR READWRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:STANDARD</td>
<td>Sets or queries the ddr readwrite search standard as DDR3 or LPDDR3.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRREADWRITE:STROBESource</td>
<td>Sets or queries the DDR readwrite strobe source when the search type is DDR READWRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRWRITE:DATARate</td>
<td>Sets or queries the DDR write search data rate for DDR3 and LPDDR3 standards of the specified search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRWRITE:DATASource</td>
<td>Sets or queries the DDR write data source when the search type is DDR Write.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRWRITE:HYSteresis</td>
<td>Sets or queries the DDR write hysteresis reference level value when the search type is DDR WRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRWRITE:MARgin</td>
<td>Sets or queries the DDR write margin reference level value, when the search type is DDR WRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRWRITE:POSTAMBLE:LENGTH</td>
<td>Sets or queries the DDR write postamble length when the search type is DDR WRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRWRITE:PREAMBLE:TYPE</td>
<td>Sets or queries the DDR write preamble type when the search type is DDR WRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH</td>
<td>Sets or queries the DDR write data high reference level value, when the search type is DDR WRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW</td>
<td>Sets or queries the DDR write data low reference level value, when the search type is DDR WRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRWRITE:REFLevel:DATA:STROBE:MID</td>
<td>Sets or queries the DDR write data mid reference level value, when the search type is DDR WRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGh</td>
<td>Sets or queries the DDR write reference level strobe high value, when the search type is DDR WRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW</td>
<td>Sets or queries the DDR write reference level strobe low value, when the search type is DDR WRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRWRITE:REFLEVELMode</td>
<td>Sets or queries the DDR write reference level mode to auto or manual, when the search type is DDR WRITE.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRWRITE:STANDARD</td>
<td>Sets or queries the DDR write search standard as DDR3 or LPDDR3.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRWRITE:STROBESource</td>
<td>Sets or queries the DDR write strobe source when the search type is DDR Write.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:EDGE:SLOpe</td>
<td>Sets or queries the slope for an edge trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:EDGE:SOURce</td>
<td>Sets or queries the source waveform for an edge trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:FUNCTion</td>
<td>Sets or queries the logic operator for a pattern or state trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:USEClockedge</td>
<td>Sets whether or not Logic search uses a clock source.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:RUNT:WHen</td>
<td>Sets or queries the condition setting for a runt trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:RUNT:WIDTH</td>
<td>Sets or queries the width setting for a runt trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SETHold:CLOCk:EDGE</td>
<td>Sets or queries the clock slope setting for a setup/hold trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SETHold:CLOCk:SOURce</td>
<td>Sets or queries the clock source setting for a setup/hold trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SETHold:HOLDTime</td>
<td>Sets or queries the hold time setting for a setup/hold trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SETHold:SETTime</td>
<td>Sets or queries the setup time setting for a setup/hold trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:TIMEOut:TIME</td>
<td>Sets or queries the time setting for a timeout trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:TRANSition:DELTATime</td>
<td>Sets or queries the transition time setting for a transition trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:TRANSition:WHen</td>
<td>Sets or queries the condition setting for a transition trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:TYPe</td>
<td>Sets or queries the trigger type setting for a search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:WINdow:WIDTH</td>
<td>Sets or queries the width setting for a window search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:LIN:CONDITION</td>
<td>Sets or queries the LIN bus search trigger condition.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:USB:DATA:QUALifier</td>
<td>Sets or queries the qualifier to be used when searching on a USB bus signal.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:EDGE:THReshold</td>
<td>Sets or queries the source threshold level for an edge trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:CLOCk:THReshold</td>
<td>Sets or queries the logic clock threshold for a logic trigger search to determine where to place a mark.</td>
</tr>
</tbody>
</table>
Table 2-38: Search and Mark commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:DELTatime</td>
<td>Specifies the Logic search delta time value.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:INPUT:CLOCK:SOUrce</td>
<td>Sets or queries the channel to use as the clock source for logic trigger.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:LEVel:CH&lt;x&gt;</td>
<td>Sets or queries the voltage level to use for logic trigger search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:LEVel:MATH&lt;x&gt;</td>
<td>Sets the voltage level to use for logic trigger search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:LEVel:REF&lt;x&gt;</td>
<td>Sets the voltage level to use for logic trigger search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:LOGICPattern:CH&lt;x&gt;</td>
<td>Sets or queries the conditions used for generating an A logic pattern.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:LOGICPattern:CH&lt;x&gt;_D&lt;x&gt;</td>
<td>Sets or queries the conditions used for generating an A logic pattern.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:LOGICPattern:MATH&lt;x&gt;</td>
<td>Sets or queries the conditions used for generating an A logic pattern.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:LOGICPattern:REF&lt;x&gt;</td>
<td>Sets or queries the conditions used for generating an A logic pattern.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:POLarity</td>
<td>Sets or queries the polarity for the clock channel when Use Clock Edge is set to Yes for Logic search type.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:WHEn</td>
<td>Sets or queries the condition for generating an A or B logic search with respect to the defined input pattern.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:PULSEWidth:HIGHLimit</td>
<td>Specifies the upper limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:PULSEWidth:LOGICQUALification</td>
<td>Specifies whether or not to use logic qualification for a pulse width search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:PULSEWidth:LOWLimit</td>
<td>Specifies the lower limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:PULSEWidth:SOUrce</td>
<td>Sets and queries the source for the pulsewidth search input.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:PULSEWidth:THReshold</td>
<td>Sets or queries the source threshold level for a pulse width trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:PULSEWidth:WHEn</td>
<td>Specifies to search for a pulse with a specified width.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:RUNT:LOGICQUALification</td>
<td>Specifies whether or not to use logic qualification for a runt search.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:RUNT:POLarity</td>
<td>Specifies the polarity for the runt search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:RUNT:THReshold:HIGH</td>
<td>Sets or queries the source threshold HIGH level for a runt trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:RUNT:THReshold:LOW</td>
<td>Sets or queries the source threshold LOW level for a runt trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SetHold:CLOCk:THReshold</td>
<td>Sets or queries the clock threshold setting for a setup/hold trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SetHold:LEVel:CH&lt;x&gt;</td>
<td>Sets or queries the voltage level to use for setup &amp; hold trigger search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SetHold:LEVel:MATH&lt;x&gt;</td>
<td>Sets or queries the voltage level to use for setup &amp; hold trigger search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SetHold:LEVel:REF&lt;x&gt;</td>
<td>Sets or queries the voltage level to use for setup &amp; hold trigger search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SetHold:LOGICPattern:CH&lt;x&gt;</td>
<td>Sets or queries the conditions used for generating an A logic pattern.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SetHold:LOGICPattern:CH&lt;x&gt;_D&lt;x&gt;</td>
<td>Sets or queries the conditions used for generating an A logic pattern.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SetHold:LOGICPattern:MATH&lt;x&gt;</td>
<td>Sets or queries the conditions used for generating an A logic pattern.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SetHold:LOGICPattern:REF&lt;x&gt;</td>
<td>Sets and returns the conditions used for generating an A logic pattern.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:STATE</td>
<td>Sets or queries the enabled state of the search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:STOPAcq</td>
<td>Sets or queries whether acquisitions are stopped when a search hit is found.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:TIMEOut:LOGICQUALification</td>
<td>Sets whether or not to use logic qualification for a timeout search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:TIMEOut:POLarity</td>
<td>Sets or queries the polarity to be used for a Timeout search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:TIMEOut:THReshold</td>
<td>Sets or queries the source threshold level for a timeout trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:TRANSition:LOGICQUALification</td>
<td>Specifies whether or not to use logic qualification for a transition search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:TRANSition:POLarity</td>
<td>Specifies the polarity for the transition search.</td>
</tr>
</tbody>
</table>
Table 2-38: Search and Mark commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:TRANSition:SOUrce</td>
<td>Sets and queries the source for the transition search input.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:TRANSition:THReshold:HIGH</td>
<td>Sets or queries the source threshold HIGH level for a transition trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:TRANSition:THReshold:LOW</td>
<td>Sets or queries the source threshold LOW level for a transition trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:WINdow:CROSSIng</td>
<td>Sets or queries the window trigger threshold crossing of the selected trigger Source.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:WINdow:LOGICQUALification</td>
<td>Specifies or queries whether or not to use logic qualification for a window search.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:WINdow:POLarity</td>
<td>Sets or queries the window trigger threshold crossing of the selected trigger Source.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:WINdow:SOUrce</td>
<td>Sets and queries the source for the window search input.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:WINdow:THReshold:HIGH</td>
<td>Sets or queries the source threshold HIGH level for a transition trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:WINdow:THReshold:LOW</td>
<td>Sets or queries the source threshold LOW level for a transition trigger search to determine where to place a mark.</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:WINdow:WHEn</td>
<td>Sets or queries the window search event.</td>
</tr>
<tr>
<td>SEARCH:SELected</td>
<td>Sets or queries the selected search.</td>
</tr>
<tr>
<td>SEARCHTABLE</td>
<td>This command adds or deletes a new search event table in an Option 5-WIN (Microsoft Windows 10 OS) TekExpress compliance testing application.</td>
</tr>
</tbody>
</table>
## Self Test command group

The Self Test commands control the selection and execution of diagnostic tests.

### Table 2-39: Self Test commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIAg:LOOP:OPTion</td>
<td>Sets or queries the type of looping desired.</td>
</tr>
<tr>
<td>DIAg:LOOP:OPTion:NTIMes</td>
<td>Sets or queries how many loops to run.</td>
</tr>
<tr>
<td>DIAg:LOOP:STOP</td>
<td>Stops diagnostics looping.</td>
</tr>
<tr>
<td>DIAg:MODE</td>
<td>Sets or queries the diagnostics mode.</td>
</tr>
<tr>
<td>DIAg:RESULT?</td>
<td>Returns the diagnostics results.</td>
</tr>
<tr>
<td>DIAg:RESULT:FLAg?</td>
<td>Returns the status of the diagnostics (single area).</td>
</tr>
<tr>
<td>DIAg:RESULT:LOG?</td>
<td>Returns the status of the diagnostic area.</td>
</tr>
<tr>
<td>DIAg:SELect</td>
<td>Selects or queries an available diagnostic area.</td>
</tr>
<tr>
<td>TOUCHSCReen:CALibrate</td>
<td>Starts the touchscreen calibration procedure.</td>
</tr>
</tbody>
</table>
Status and Error command group

Use the commands in the Status and Error command Group to determine the status of the instrument and control events.

Several commands and queries used with the instrument are common to all devices. The IEEE Std 488.2-1987 defines these commands and queries. The common commands begin with an asterisk (*) character.

Table 2-40: Status and Error commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLEv?</td>
<td>Returns all events and their messages.</td>
</tr>
<tr>
<td>BUSY?</td>
<td>Returns instrument status.</td>
</tr>
<tr>
<td>*CLS</td>
<td>Clears status.</td>
</tr>
<tr>
<td>DESE</td>
<td>Sets or queries the bits in the Device Event Status Enable Register.</td>
</tr>
<tr>
<td>*ESE</td>
<td>Sets or queries the bits in the Event Status Enable Register.</td>
</tr>
<tr>
<td>*ESR?</td>
<td>Returns the contents of the Standard Event Status Register.</td>
</tr>
<tr>
<td>EVENT?</td>
<td>Returns event code from the event queue.</td>
</tr>
<tr>
<td>EVMsg?</td>
<td>Returns event code, message from the event queue.</td>
</tr>
<tr>
<td>EVQty?</td>
<td>Returns number of events that are enabled in the queue.</td>
</tr>
<tr>
<td>*OPC</td>
<td>Generates the operation complete message in the standard event status register when all pending operations are finished. Or returns “1” when all current operations are finished.</td>
</tr>
<tr>
<td>*OPT?</td>
<td>Returns a comma separated list of installed options as an arbitrary ASCII string (no quotes).</td>
</tr>
<tr>
<td>*PSC</td>
<td>Sets or queries the power on status flag.</td>
</tr>
<tr>
<td>*PUD</td>
<td>Sets or queries a string of protected user data.</td>
</tr>
<tr>
<td>*RST</td>
<td>Resets the instrument to factory default settings.</td>
</tr>
<tr>
<td>*SRE</td>
<td>Sets or queries the bits in the Service Request Enable Register.</td>
</tr>
<tr>
<td>*STB?</td>
<td>Returns the contents of the Status Byte Register.</td>
</tr>
<tr>
<td>*WAI</td>
<td>Prevents the instrument from executing further commands until all pending operations finish.</td>
</tr>
</tbody>
</table>
Command groups

Trigger command group

Use the commands in the Trigger Command Group to control all aspects of triggering for the instrument.

There are two triggers: A and B. Where appropriate, the command set has parallel constructions for each trigger.

You can set the A or B triggers to edge mode. Edge triggering lets you display a waveform at or near the point where the signal passes through a voltage level of your choosing.

You can also set A or B triggers to pulse or logic modes. With pulse triggering, the instrument triggers whenever it detects a pulse of a certain width or height. Logic triggering lets you logically combine the signals on one or more channels. The instrument then triggers when it detects a certain combination of signal levels. The trigger types of Pulse Width, Timeout, Runt, Window, and Rise/Fall Time can be further qualified by a logic pattern. This is referred to as logic qualification.

Table 2-41: Trigger commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger</td>
<td>Forces a trigger event to occur or returns current trigger parameters for the instrument.</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:BUS:B&lt;x&gt;:ARINC429A:CONDition</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:BUS:B&lt;x&gt;:ARINC429A:ERRTYPE</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:BUS:B&lt;x&gt;:ARINC429A:LABel:QUALifier</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:BUS:B&lt;x&gt;:ARINC429A:LABel:VALue</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:BUS:B&lt;x&gt;:ARINC429A:LABel:HIVALue</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:BUS:B&lt;x&gt;:ARINC429A:SDI:VALue</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:BUS:B&lt;x&gt;:ARINC429A:SSM:VALue</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:BUS:B&lt;x&gt;:ARINC429A:DATa:QUALifier</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:BUS:B&lt;x&gt;:ARINC429A:DATa:VALue</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:BUS:B&lt;x&gt;:ARINC429A:DATa:HIVALue</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:BUS:B&lt;x&gt;:AUDio:CONDition</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:BUS:B&lt;x&gt;:AUDio:DATa:HIVALue</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:AUDio:DATa:HM</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:AUDio:DATa:OFF</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:AUDio:DATa:QUALi</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:AUDio:DATa:TM</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:AUDio:DATa:VA</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:AUDio:DATa:WO</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:CAN:CONDition</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:CAN:DATa:DIRection</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:CAN:DATa:OFFSet</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:CAN:DATa:QUALi</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:CAN:DATa:SIZe</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:CAN:DATa:VA</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:CAN:ERRType</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:CAN:FD:BRSBit</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:CAN:FD:ESIBit</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:CAN:FRAMEtype</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:CAN:IDENTifier:MOD</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:CAN:IDENTifier:VALu</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:CONDition</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:DATa:HI</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:DATa:OFF</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:DATa:QUALi</td>
</tr>
</tbody>
</table>
### Table 2-41: Trigger commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:DATa:SIZE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:DATa:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:IPHeader:DESTinationaddr:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:IPHeader:PROTocol:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:IPHeader:SOUrceaddr:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:MAC:ADDRess:DESTination:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:MAC:ADDRess:SOUrce:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:MAC:LENgth:HIVALue</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:MAC:LENgth:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:QTAG:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:TCPHeader:ACKnum:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:TCPHeader:DESTinationport:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:TCPHeader:SEQnum:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:ETHERnet:TCPHeader:SOUrceport:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:CONDition</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:CYCLEcount:HIVALue</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:CYCLEcount:QUALifier</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:CYCLEcount:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:DATa:HIVALue</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:DATa:OFFSet</td>
</tr>
</tbody>
</table>
Table 2-41: Trigger commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:DATa:QUALifier</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:DATa:SIZE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:DATa:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:EOFTYPE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:ERTYPE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:FRAMEID:HIVALue</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:FRAMEID:QUALifier</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:FRAMEID:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:FRAMEType</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:HEADER:CRC</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:HEADER:CYCLeCount</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:HEADER:FRAMEID</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:HEADER:INDBits</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:FLEXray:HEADER:PAYLength</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:I2C:ADDReSS:MODe</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:I2C:ADDReSS:VALue</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:I2C:CONDition</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:I2C:DATa:DIRection</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:I2C:DATa:SIZE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:I2C:DATa:VALue</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:LIN:CONDition</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:LIN:DATa:HIVALue</td>
</tr>
</tbody>
</table>
Table 2-41: Trigger commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:LIN:DATA:QUALifier</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:LIN:DATA:Size</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:LIN:DATA:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:LIN:ERTYPE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:LIN:IDENTifier:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:CONDITION</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:ERTYPE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:COMMAND:TRBit</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:COMMAND:PARity</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:COMMAND:COUNt</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:COMMAND:SUBADdress</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:COMMAND:ADDRess:QUALifier</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:COMMAND:ADDRess:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:COMMAND:ADDRess:HIGHValue</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:DATA:PARity</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:DATA:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:STATus:PARity</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:STATus:ADDRess:QUALifier</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:STATus:ADDRess:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:STATus:ADDRess:HIGHValue</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:STATus:BIT:ME</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:STATus:BIT:INSTR</td>
</tr>
</tbody>
</table>
### Table 2-41: Trigger commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:STATus:BIT:SRQ</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:STATus:BIT:BCR</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:STATus:BIT:BUSY</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:STATus:BIT:SUBSF</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:STATus:BIT:DBCA</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:STATus:BIT:TF</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:TIMe:QUALifier</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:TIMe:LESSLimit</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:MIL1553B:TIMe:MORELimit</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:PARallel:DATa:VALue</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:RS232C:DATa:SiZe</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:RS232C:DATa:VALue</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:CONDition</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:ERRType:CRC</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:FAST:CHAN1A:HIVALue</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:FAST:CHAN1A:QUALifier</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:FAST:CHAN1A:VALue</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:FAST:CHAN2B:HIVALue</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:FAST:CHAN2B:QUALifier</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:FAST:CHAN2B:VALue</td>
</tr>
</tbody>
</table>
## Table 2-41: Trigger commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:FAST:COUNTer:HIVALue`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:FAST:COUNTer:QUALifier`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:FAST:COUNTer:VALue`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:FAST:INVERTNIBble:VALue`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:PAUSE:QUALifier`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:SLOW:DATA:HIVALue`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:SLOW:DATA:QUALifier`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:SLOW:DATA:VALue`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SENT:SLOW:IDENTifier:VALue`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SPI:CONDition`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SPI:DATa:SiZe`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SPI:DATa:VALue`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SPI:CONDition`</td>
</tr>
<tr>
<td><code>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:SPMI:DATa:SiZe</code></td>
<td>This command sets or queries the length of the data string, in bytes, to be used when triggering on an SPMI bus signal.</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SPMI:DATa:VALue`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SPMI:MASTERADDRess:VALue`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SPMI:NORESPonse`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SPMI:REGISTERADDRess:VALue`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:SPMI:SLAVEADDRess:VALue`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:ADDRess:HIVALue`</td>
</tr>
<tr>
<td>`TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:ADDRess:VALue`</td>
</tr>
</tbody>
</table>
Table 2-41: Trigger commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:CONDition</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:DATA:HIVALue</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:DATA:OFFSET</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:DATA:QUALifier</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:DATA:SIZE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:DATA:TYPE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:DATA:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:ENDPoint:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:ERRType</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:HANdSHAKEType</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:SOFFRAMENUMber</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:SPECIALType</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:SPLIT:ET:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:SPLIT:HUB:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:SPLIT:PORT:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:SPLIT:SC:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:SPLIT:SE:VALUE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:B&lt;x&gt;:USB:TOKENType</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:BUS:SOURce</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:EDGE:COUPling</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:EDGE:SLOpe</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:EDGE:SOURce</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:LEVel:CH&lt;x&gt;</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:LOGIC:DELTatime</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:LOGIC:FUNCTION</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:LOGIC:INPut:CLOCK:SOUrce</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:LOGIC:POLarity</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:LOGIC:USECLOckedge</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:LOGIC:WHEn</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:LOGICPattern: {CH&lt;x&gt;</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:LOWerthreshold:CH&lt;x&gt;</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:PULSEWidth:LOGICQUALification</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:PULSEWidth:HIGHLimit</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:PULSEWidth:LOWLimit</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:PULSEWidth:POLarity</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:PULSEWidth:SOUrce</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:PULSEWidth:WHEn</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:RUNT:LOGICQUALification</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:RUNT:POLarity</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:RUNT:SOUrce</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:RUNT:WHEn</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:RUNT:WIDth</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:SETHold:CLOCk:EDGE</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:SETHold:CLOCk:SOUrce</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:SETHold:HOLDTime</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:SETHold:SETTime</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:SETHOLDLogicval: {CH&lt;x&gt;</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:TIMEOut:LOGICQUALification</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:TIMEOut:POLarity</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:TIMEOut:SOUrce</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:TIMEOut:TIME</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:TRANsition:DELTatime</td>
</tr>
<tr>
<td>TRIGger:[A</td>
<td>B]:TRANsition:POLarity</td>
</tr>
</tbody>
</table>
### Table 2-41: Trigger commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger:{A</td>
<td>B}:TRANsition:SOUrce</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:TRANsition:WHen</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:TRANsition:LOGICQUALification</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:TYPE</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:UPPerthreshold:CH&lt;x&gt;</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:WINdow:CROSSing</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:WINdow:LOGICQUALification</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:WINdow:SOUrce</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:WINdow:WHen</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:WINdow:WIDth</td>
</tr>
<tr>
<td>TRIGger:{A</td>
<td>B}:RESET</td>
</tr>
<tr>
<td>TRIGger:A:HOLDoff:BY</td>
<td>Sets or queries the type of holdoff for the A trigger.</td>
</tr>
<tr>
<td>TRIGger:A:HOLDoff:TIMe</td>
<td>Sets or queries the A trigger holdoff time.</td>
</tr>
<tr>
<td>TRIGger:A:LOGICQUALification</td>
<td>Sets or queries the type of logic qualification to perform.</td>
</tr>
<tr>
<td>TRIGger:A:MODe</td>
<td>Sets or queries the A trigger mode.</td>
</tr>
<tr>
<td>TRIGger:AUXLevel</td>
<td>Sets or queries the Auxiliary Input voltage level to use for an edge trigger.</td>
</tr>
<tr>
<td>TRIGger:B:BY</td>
<td>Sets or queries B trigger time or event qualifiers.</td>
</tr>
<tr>
<td>TRIGger:B:EVENTS:COUNt</td>
<td>Sets or queries the number of events that must occur before the B trigger occurs.</td>
</tr>
<tr>
<td>TRIGger:B:RESET</td>
<td>Sets the B reset trigger level to 50%.</td>
</tr>
<tr>
<td>TRIGger:B:RESET:EDGE:COUpling</td>
<td>Sets or queries the trigger coupling for a sequential edge trigger reset when the Source is set to an analog channel.</td>
</tr>
<tr>
<td>TRIGger:B:RESET:EDGE:LEVel</td>
<td>Sets the voltage level to use for an Edge Reset trigger when triggering on an analog channel waveform.</td>
</tr>
<tr>
<td>TRIGger:B:RESET:EDGE:SLOpe</td>
<td>Sets or queries the trigger slope for a sequential edge trigger reset.</td>
</tr>
<tr>
<td>TRIGger:B:RESET:EDGE:SOUrce</td>
<td>Sets or queries the trigger source for the A→B sequential edge trigger reset feature.</td>
</tr>
<tr>
<td>TRIGger:B:RESET:TIMEOut:TIMe</td>
<td>Sets or queries the reset timer for a sequential time out trigger reset.</td>
</tr>
<tr>
<td>TRIGger:B:RESET:TYPE</td>
<td>Sets or queries the type of A→B sequential trigger reset.</td>
</tr>
<tr>
<td>TRIGger:B:STATE</td>
<td>Returns the current state of the triggering system.</td>
</tr>
<tr>
<td>TRIGger:B:TIMe</td>
<td>Sets or queries the B trigger delay time.</td>
</tr>
<tr>
<td>TRIGger:STATE?</td>
<td>Returns the current state of the triggering system.</td>
</tr>
<tr>
<td>VISual:AREA&lt;x&gt;:ASPEctratio</td>
<td>Sets or queries whether the aspect ratio of the specified Visual Trigger area is locked.</td>
</tr>
<tr>
<td>VISual:AREA&lt;x&gt;:FLIP:HORizontal</td>
<td>Flips the specified Visual Trigger area horizontally around its center point.</td>
</tr>
<tr>
<td>VISual:AREA&lt;x&gt;:FLIP:VERTical</td>
<td>Flips the specified Visual Trigger area vertically around its center point.</td>
</tr>
<tr>
<td>VISual:AREA&lt;x&gt;:HEIGHT</td>
<td>Sets or queries the height of the specified Visual Trigger area.</td>
</tr>
<tr>
<td>VISual:AREA&lt;x&gt;:HITType</td>
<td>Sets or queries the area hit logic true condition for the specified Visual Trigger area.</td>
</tr>
</tbody>
</table>
### Table 2-41: Trigger commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VISual:AREA&lt;x&gt;:RESET</td>
<td>Sets the specified Visual Trigger area shape to a default-sized triangle.</td>
</tr>
<tr>
<td>VISual:AREA&lt;x&gt;:ROTAtion</td>
<td>Sets or queries the rotation angle of the specified Visual Trigger area.</td>
</tr>
<tr>
<td>VISual:AREA&lt;x&gt;:SHAPE</td>
<td>Sets or queries the current shape of the area.</td>
</tr>
<tr>
<td>VISual:AREA&lt;x&gt;:SOUrce</td>
<td>Sets or queries the signal source for the specified Visual Trigger area.</td>
</tr>
<tr>
<td>VISual:AREA&lt;x&gt;:VERTICES</td>
<td>Sets or queries the X and Y vertex coordinate values for all vertices of the specified Visual Trigger area.</td>
</tr>
<tr>
<td>VISual:AREA&lt;x&gt;:WIDTH</td>
<td>Sets or queries the width of the specified Visual Trigger area.</td>
</tr>
<tr>
<td>VISual:AREA&lt;x&gt;:XPOSition</td>
<td>Sets or queries the horizontal (X-axis) center position of the specified Visual Trigger area.</td>
</tr>
<tr>
<td>VISual:AREA&lt;x&gt;:YPOSition</td>
<td>Sets or queries the vertical (Y-axis) center position of the specified Visual Trigger area.</td>
</tr>
<tr>
<td>VISual:DELETEALL</td>
<td>Deletes all Visual Trigger areas.</td>
</tr>
<tr>
<td>VISual:ENAble</td>
<td>Sets or queries the status (on or off) of the Visual Trigger mode.</td>
</tr>
<tr>
<td>VISual:EQUation</td>
<td>Sets or queries the Visual Trigger area combination logic equation.</td>
</tr>
<tr>
<td>VISual:SHOWAReas</td>
<td>Shows or hides all Visual Trigger areas.</td>
</tr>
<tr>
<td>VISual:SHOWCRiteria</td>
<td>Sets or queries display of the area names and hit criteria for all visual trigger areas.</td>
</tr>
<tr>
<td>VISual:SHOWEQuation</td>
<td>Shows or hides the Visual Trigger area combination logic equation.</td>
</tr>
</tbody>
</table>
Vertical command group

Use the commands in the Vertical Command Group to control the vertical setup of all live (channel) waveforms for acquisition and to control the display of channel, reference, and math waveforms. Analog channels are available when analog probes are attached to FlexChannel inputs.

You can replace VOLts with SCAle in the vertical commands. This provides program compatibility with earlier models of Tektronix instruments.

Table 2-42: Vertical commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH&lt;x&gt;?</td>
<td>Returns vertical parameters for the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:BANwidth</td>
<td>Sets or queries the bandwidth of the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:CLIPping?</td>
<td>Queries whether the specified channel’s input signal is clipping (exceeding) the channel vertical scale setting.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:COUPling</td>
<td>Sets or queries the coupling setting for the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:DESKeW</td>
<td>Sets or queries the deskeW time for the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:LABel:COlor</td>
<td>Sets or queries the color of the specified channel label.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:LABel:FONT:BOLD</td>
<td>Sets or queries the bold state of the specified channel label.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:LABel:FONT:ITALic</td>
<td>Sets or queries the italic state of the specified channel label.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:LABel:FONT:SIZE</td>
<td>Sets or queries the font size of the specified channel label.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:LABel:FONT:TYPE</td>
<td>Sets or queries the font type of the specified channel label.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:LABel:FONT:UNDERline</td>
<td>Sets or queries the underline state of the specified channel label.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:LABel:NAMe</td>
<td>Defines or queries the label for the channel waveform.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:LABel:XPOS</td>
<td>Sets or queries the X display coordinate for the channel waveform label.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:LABel:YPOS</td>
<td>Sets or queries the Y display coordinate for the channel waveform label.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:OFFSet</td>
<td>Sets or queries the channel offset.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:POSition</td>
<td>Sets or queries the vertical position for the specified analog channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe?</td>
<td>Returns all information concerning the probe that is attached to the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:AUTOZero</td>
<td>Autozeros the probe attached to the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:DEGAUSS</td>
<td>Starts a degauss cycle of the probe attached to the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:DEGAUSS:STATE?</td>
<td>Queries whether the probe attached to the specified channel requires a degauss operation.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:FORCEDRange</td>
<td>Sets the attached probe to its default range, or it queries the default range of the probe.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:GAIN?</td>
<td>Returns the gain of the probe that is attached to the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:ID?</td>
<td>Returns the type and serial number of the probe that is attached to the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:ID:SERnumber?</td>
<td>Returns the serial number of the probe that is attached to the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:ID:TYPE?</td>
<td>Returns the type of probe that is attached to the specified channel.</td>
</tr>
</tbody>
</table>
### Table 2-42: Vertical commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH&lt;x&gt;:PRObe:INPUTMode</td>
<td>Sets or queries the input mode of the probe that is attached to the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:INPUTMode:AOFFSet</td>
<td>Sets or queries the A mode offset value of the probe that is attached to the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:INPUTMode:BOFFSet</td>
<td>Sets or queries the B mode offset value of the probe that is attached to the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:INPUTMode:COFFSet</td>
<td>Sets or queries the common mode offset value of the probe that is attached to the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:INPUTMode:DOFFSet</td>
<td>Sets or queries the differential mode offset value of the probe that is attached to the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:RESistance?</td>
<td>Queries the resistance of the probe that is attached to the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:SET</td>
<td>Sets or queries aspects of probe accessory user interfaces, for example probe attenuation factors.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:UNIts?</td>
<td>Returns the units of the probe that is attached to the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PROBECal?</td>
<td>Returns the probe calibration status.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PROBEFunc:EXTAtten</td>
<td>Sets the attenuation value for the specified channel to the specified scale factor. Or queries the user-specified attenuation.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PROBEFunc:EXTDBatten</td>
<td>Sets the attenuation value for the specified channel to the specified value in decibels. Or queries the user-specified attenuation in decibels.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PROBEFunc:EXTUnits</td>
<td>Sets or queries the alternate unit for the external attenuator of the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PROBEFunc:EXTUnits:STATE</td>
<td>Sets or queries the custom units enable state for the specified channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:SCALE</td>
<td>Sets or returns the vertical scale for the specified analog channel.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:TERmination</td>
<td>Sets channel input termination.</td>
</tr>
<tr>
<td>CH&lt;x&gt;:VTERm:BIAS</td>
<td>Sets or queries the termination voltage for the specified channel (if control is available).</td>
</tr>
<tr>
<td>CONFIGuration:ANALOG:BANDwidth?</td>
<td>Queries the maximum licensed bandwidth of the instrument.</td>
</tr>
<tr>
<td>REF:ADDNew</td>
<td>Adds the specified reference. Argument is of the form &quot;REF&lt;NR1&gt; &quot;., where NR1 ≥ 1.</td>
</tr>
<tr>
<td>REF:DELETE</td>
<td>Deletes the specified reference.</td>
</tr>
<tr>
<td>REF:REF&lt;x&gt;:DESKew</td>
<td>Sets or queries the deskew time for the specified reference.</td>
</tr>
<tr>
<td>REF:REF&lt;x&gt;:LABel:COlor</td>
<td>Sets or queries the color of the specified ref label.</td>
</tr>
<tr>
<td>REF:REF&lt;x&gt;:LABel:FONT:BOLD</td>
<td>Sets or queries the bold state of the specified reference label.</td>
</tr>
<tr>
<td>REF:REF&lt;x&gt;:LABel:FONT:ITALic</td>
<td>Sets or queries the italic state of the specified reference label.</td>
</tr>
<tr>
<td>REF:REF&lt;x&gt;:LABel:FONT:SIZE</td>
<td>Sets or queries the font size of the specified reference label.</td>
</tr>
<tr>
<td>REF:REF&lt;x&gt;:LABel:FONT:TYPE</td>
<td>Sets or queries the font type of the specified reference label.</td>
</tr>
<tr>
<td>REF:REF&lt;x&gt;:LABel:FONT:UNDERline</td>
<td>Sets or queries the underline state of the specified reference label.</td>
</tr>
</tbody>
</table>
### Table 2-42: Vertical commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REF:REF&lt;x&gt;:LABel:NAME</td>
<td>Sets or queries the label of the designated waveform.</td>
</tr>
<tr>
<td>REF:REF&lt;x&gt;:LABel:XPOS</td>
<td>Sets or queries the position of the reference waveform label on the X axis.</td>
</tr>
<tr>
<td>REF:REF&lt;x&gt;:LABel:YPOS</td>
<td>Sets or queries the position of the reference waveform label on the Y axis.</td>
</tr>
<tr>
<td>REF:LIST?</td>
<td>Lists all currently defined references.</td>
</tr>
<tr>
<td>REF:REF&lt;x&gt;:SOURce</td>
<td>Sets or queries the filename used by the given reference.</td>
</tr>
<tr>
<td>VERTical:DESKEW:FROM:CUSTOMPROPagation</td>
<td>Sets or queries a target (FROM) delay that you can specify when the propagation delay of the target (FROM) probe used for deskew cannot be detected automatically.</td>
</tr>
<tr>
<td>VERTical:DESKEW:STATIC</td>
<td>Sets or queries the target channel for performing channel-to-channel deskew adjustment.</td>
</tr>
<tr>
<td>VERTical:DESKEW:TO:CUSTOMPROPagation</td>
<td>Sets or queries a target (TO) delay that can be specified by the user when the propagation delay of the target (TO) probe used for deskew cannot be detected automatically.</td>
</tr>
<tr>
<td>VERTical:DESKEW:FROM:SOURce</td>
<td>Sets or queries the source channel for performing channel-to-channel deskew adjustment.</td>
</tr>
<tr>
<td>VERTical:DESKEW:TO:SOURce</td>
<td>Sets or queries the target channel for performing channel-to-channel deskew adjustment.</td>
</tr>
</tbody>
</table>
Waveform Transfer command group

Use the commands in the Waveform Transfer Command Group to transfer waveform data points from the instrument. Waveform data points are a collection of values that define a waveform. One data value usually represents one data point in the waveform record. When working with envelope waveforms, each data value is either the minimum or maximum of a min/max pair.

Before you transfer waveform data, you must specify the data format, record length, and waveform source.

**Data formats.** Acquired waveform data uses eight or more bits to represent each data point. The number of bits used depends on the acquisition mode specified when you acquired the data. Data acquired in SAMple or ENvelope mode uses eight bits per waveform data point. Data acquired in AVERAGE mode uses up to 14 bits per point.

The instrument can transfer waveform data in either ASCII or binary format. You specify the format with the DATa:ENCdg command. The instrument uses signed, 4 byte integers and floating point values; it does not support unsigned floating point values.

**ASCII data** is represented by signed integer or floating point values. An example ASCII waveform data string can look like this:

```
CURVE<space>-110,-109,-110,-110,-109,-107,-109,-107,-106,-105,-103,-100,-97,-90,-84,-80
```

Use ASCII to obtain more readable and easier to format output than binary. However, ASCII can require more bytes to send the same values than it does with binary. This can reduce transmission speeds.

**Binary data** can be represented by signed integer or floating point values. The range of the values depends on the byte width specified. When the byte width is one, signed integer data ranges from -128 to 127, and positive integer values range from 0 to 255. When the byte width is two, the values range from -32768 to 32767. When a MATH (or REF that came from a MATH) is used, 32-bit floating point values are used that are four bytes in width.
The defined binary formats specify the order in which the bytes are transferred. The following are the four binary formats:

- **RIBinary** specifies signed integer data-point representation with the most significant byte transferred first.
- **SRIBinary** is the same as RIBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.
- **RFBinary** specifies floating point data-point representation with the most significant byte transferred first.
- **SRFBinary** is the same as RFBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.

**Waveform data and record lengths.** You can transfer multiple points for each waveform record. You can transfer a portion of the waveform or you can transfer the entire record. You can use the `DATA:STARt` and `DATA:STOP` commands to specify the first and last data points of the waveform record.

When transferring data from the instrument, you must specify the first and last data points in the waveform record. Setting `DATA:STARt` to 1 and `DATA:STOP` to the record length will always return the entire waveform.

**Waveform data locations and memory allocation.** The `DATA:SOURce` command specifies the waveform source when transferring a waveform from the instrument.

**Waveform preamble.** Each waveform that you transfer has an associated waveform preamble that contains information such as the horizontal scale, the vertical scale, and other settings in effect when the waveform was created. Refer to the individual `WFMOutpre?` commands for more information.

**Scaling waveform data.** Once you transfer the waveform data to the controller, you can convert the data points into voltage values for analysis using information from the waveform preamble.
Transferring waveform data from the instrument.

You can transfer waveforms from the instrument to an external controller using the following sequence:

1. Select the waveform source(s) using \texttt{DATa:SOUrce}.
2. Specify the waveform data format using \texttt{DATa:ENCdg}.
3. Specify the number of bytes per data point using \texttt{WFMOutpre:BYT_Nr}.

\textit{NOTE.} \textit{MATH} waveforms (and \textit{REF} waveforms that came from a \textit{MATH}) are always set to four bytes.

4. Specify the portion of the waveform that you want to transfer using \texttt{DATa:STARt} and \texttt{DATa:STOP}.
5. Transfer waveform preamble information using \texttt{WFMOutpre}.
6. Transfer waveform data from the instrument using \texttt{CURVe?}.

Table 2-43: Waveform Transfer commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{CURVe}</td>
<td>The query format transfers waveform data from instrument specified by the \texttt{DATa:SOUrce} command.</td>
</tr>
<tr>
<td>\texttt{DATa}</td>
<td>Sets the format and location of the waveform data that is transferred with the \texttt{CURVe} Command. Or queries the format and location of the waveform data that is transferred with the \texttt{CURVe?} command.</td>
</tr>
<tr>
<td>\texttt{DATa:ENCdg}</td>
<td>Sets or queries the format of outgoing waveform data.</td>
</tr>
<tr>
<td>\texttt{DATa:SOUrce}</td>
<td>Sets or queries the location of waveform data transferred from the instrument.</td>
</tr>
<tr>
<td>\texttt{DATa:SOUrce:AVAILable?}</td>
<td>Returns a list of enumerations representing the source waveforms that are currently available for \texttt{:CURVe?} queries.</td>
</tr>
<tr>
<td>\texttt{DATa:STARt}</td>
<td>Sets or queries the starting point in waveform transfer.</td>
</tr>
<tr>
<td>\texttt{DATa:STOP}</td>
<td>Sets or queries the ending data point in waveform transfer.</td>
</tr>
<tr>
<td>\texttt{DATa:WIDth}</td>
<td>Specifies the width, in bytes per point, for waveform data transferred from the oscilloscope via the \texttt{CURVe?} query.</td>
</tr>
<tr>
<td>\texttt{WAVFrm?}</td>
<td>Returns \texttt{WFMOutpre?} and \texttt{CURVe?} data for the waveform as specified by the \texttt{DATa:SOUrce} command.</td>
</tr>
<tr>
<td>\texttt{WFMOutpre?}</td>
<td>Returns the waveform formatting data for the waveform specified by the \texttt{DATa:SOUrce} command.</td>
</tr>
<tr>
<td>\texttt{WFMOutpre:ASC_Fmt?}</td>
<td>Returns the format for ASCII data transferred from the instrument.</td>
</tr>
<tr>
<td>\texttt{WFMOutpre:BIT_Nr}</td>
<td>Sets or queries the number of bits per waveform point that outgoing waveforms contain.</td>
</tr>
<tr>
<td>\texttt{WFMOutpre:BN_Fmt}</td>
<td>Sets or queries the format of binary data for the waveform.</td>
</tr>
</tbody>
</table>
Table 2-43: Waveform Transfer commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFMOutpre:BYT_Nr</td>
<td>Sets or queries the data width for the waveform.</td>
</tr>
<tr>
<td>WFMOutpre:BYT_Or</td>
<td>Sets or queries the byte order of waveform points.</td>
</tr>
<tr>
<td>WFMOutpre:DOMain?</td>
<td>Returns the domain of the outgoing waveform.</td>
</tr>
<tr>
<td>WFMOutpre:ENCdg</td>
<td>Sets or queries the type of encoding for outgoing waveforms.</td>
</tr>
<tr>
<td>WFMOutpre:NR_Pt?</td>
<td>Returns the number of points for the waveform transmitted in response to a CURVe? query.</td>
</tr>
<tr>
<td>WFMOutpre:PT_Fmt?</td>
<td>Returns the point format for the waveform.</td>
</tr>
<tr>
<td>WFMOutpre:PT_Off?</td>
<td>Returns the trigger point relative to DATA:START for the waveform.</td>
</tr>
<tr>
<td>WFMOutpre:WFId?</td>
<td>Returns a string describing the acquisition parameters for the waveform.</td>
</tr>
<tr>
<td>WFMOutpre:XINcr?</td>
<td>Returns the horizontal sampling interval.</td>
</tr>
<tr>
<td>WFMOutpre:XUNIT?</td>
<td>Returns the horizontal units for the waveform.</td>
</tr>
<tr>
<td>WFMOutpre:YZero?</td>
<td>Returns the (sub-sample) time between the trigger sample and the occurrence of the actual waveform trigger.</td>
</tr>
<tr>
<td>WFMOutpre:YMUlt?</td>
<td>Returns the vertical scale factor per digitizing level for the waveform.</td>
</tr>
<tr>
<td>WFMOutpre:YOFF?</td>
<td>Returns the vertical offset in digitizing levels for the waveform.</td>
</tr>
<tr>
<td>WFMOutpre:YUNIT?</td>
<td>Returns the vertical units for the waveform.</td>
</tr>
<tr>
<td>WFMOutpre:YZero?</td>
<td>Returns the vertical offset for the waveform.</td>
</tr>
</tbody>
</table>
## Zoom command group

Zoom commands let you expand and position the waveform display horizontally and vertically, without changing the time base or vertical settings.

*NOTE. Zoom commands are available once a view has been added.*

### Table 2-44: Zoom commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DIsp:MATHTFFView&lt;x&gt;:ZOOM:XAXIS:FROM</code></td>
<td>Sets or queries the start of the zoom x axis in the specified plot view.</td>
</tr>
<tr>
<td><code>DIsp:MATHTFFView&lt;x&gt;:ZOOM:XAXIS:TO</code></td>
<td>Sets or queries the end of the zoom x axis in the specified plot view.</td>
</tr>
<tr>
<td><code>DIsp:MATHTFFView&lt;x&gt;:ZOOM:YAXIS:FROM</code></td>
<td>Sets or queries the start of the zoom y axis in the specified plot view.</td>
</tr>
<tr>
<td><code>DIsp:MATHTFFView&lt;x&gt;:ZOOM:YAXIS:TO</code></td>
<td>Sets or queries the end of the zoom y axis in the specified plot view.</td>
</tr>
<tr>
<td><code>DIsp:PLTView&lt;x&gt;:ZOOM:XAXIS:FROM</code></td>
<td>Sets or queries the start of the zoom x-axis in the specified plot view.</td>
</tr>
<tr>
<td><code>DIsp:PLTView&lt;x&gt;:ZOOM:XAXIS:TO</code></td>
<td>Sets or queries the end of the zoom x-axis in the specified plot view.</td>
</tr>
<tr>
<td><code>DIsp:PLTView&lt;x&gt;:ZOOM:YAXIS:FROM</code></td>
<td>Sets or queries the start of the zoom y-axis in the specified plot view.</td>
</tr>
<tr>
<td><code>DIsp:PLTView&lt;x&gt;:ZOOM:YAXIS:TO</code></td>
<td>Sets or queries the end of the zoom y-axis in the specified plot view.</td>
</tr>
<tr>
<td><code>DIsp:REFFFTView&lt;x&gt;:ZOOM:XAXIS:FROM</code></td>
<td>Sets or returns the left edge of the zoom x-axis in the specified plot view.</td>
</tr>
<tr>
<td><code>DIsp:REFFFTView&lt;x&gt;:ZOOM:XAXIS:TO</code></td>
<td>Sets or queries the right edge of the zoom x-axis in the specified plot view.</td>
</tr>
<tr>
<td><code>DIsp:REFFFTView&lt;x&gt;:ZOOM:YAXIS:FROM</code></td>
<td>Sets or queries the bottom value of the zoom y-axis in the specified plot view.</td>
</tr>
<tr>
<td><code>DIsp:REFFFTView&lt;x&gt;:ZOOM:YAXIS:TO</code></td>
<td>Sets or queries the top value of the zoom y-axis in the specified plot view.</td>
</tr>
<tr>
<td><code>DIsp:WAVEView&lt;x&gt;:ZOOM:ZOOM&lt;x&gt;:HORizontal:POSition</code></td>
<td>Sets or queries the horizontal zoom position (of the specified zoom in the specified waveview) of the zoomed waveform or zoom waveform in the display, around which the zoom waveform displays.</td>
</tr>
<tr>
<td><code>DIsp:WAVEView&lt;x&gt;:ZOOM:ZOOM&lt;x&gt;:HORizontal:SCAlE</code></td>
<td>Sets or queries the zoom horizontal scale factor, of the specified zoom in the specified waveview, in which the zoom waveform is displayed.</td>
</tr>
<tr>
<td><code>DIsp:WAVEView&lt;x&gt;:ZOOM:ZOOM&lt;x&gt;:HORizontal:WINSCAlE</code></td>
<td>Sets or queries the zoom window horizontal scale in the specified waveview.</td>
</tr>
<tr>
<td><code>DIsp:WAVEView&lt;x&gt;:ZOOM:ZOOM&lt;x&gt;:STAtE</code></td>
<td>Sets or queries the specified zoom in the specified waveview on or off.</td>
</tr>
<tr>
<td><code>DIsp:WAVEView&lt;x&gt;:ZOOM:ZOOM&lt;x&gt;:VERTical:POSition</code></td>
<td>Sets or queries the vertical position of the specified zoom in the specified waveview, in which the zoom waveform is displayed.</td>
</tr>
<tr>
<td><code>DIsp:WAVEView&lt;x&gt;:ZOOM:ZOOM&lt;x&gt;:VERTical:SCAlE</code></td>
<td>Sets or queries the zoom vertical scale factor of the specified waveform for the specified zoom in the specified waveview.</td>
</tr>
<tr>
<td><code>DIsp:WAVEView&lt;x&gt;:Zoom?</code></td>
<td>Queries the zoom parameters of the specified waveview.</td>
</tr>
<tr>
<td><code>DIsp:WAVEView&lt;x&gt;:ZOOM:ZOOM&lt;x&gt;?</code></td>
<td>Queries the zoom parameters of the specified zoom in the specified waveview.</td>
</tr>
</tbody>
</table>
Commands listed in alphabetical order

**NOTE.** Some of the following commands may not be available on your instrument model. Also, some of the following commands are only available if your instrument has the associated option installed.

**ACQuire? (Query Only)**

Queries the current acquisition state.

**Group** Acquisition

**Syntax** ACQuire?

**Examples** ACQUIRE? might return: ACQUIRE:STATE 1; NUMENV INFINITE; STOPAFTER RUNSTOP; SEQUENCE:MODE NUMACQS; NUMSEQUENCE 1; :ACQUIRE:NUMAVG 16; FASTACQ:STATE 0; PALETTE TEMPERATURE; :ACQUIRE:MODE SAMPLE; SAMPLINGMODE IT.

**ACQuire:FASTAcq:PALEtte**

Sets or queries the waveform grading for fast acquisition mode.

**Group** Acquisition

**Syntax** ACQuire:FASTAcq:PALEtte

{NORMal|TEMPerature|SPECtral|INVErted}

ACQuire:FASTAcq:PALEtte?

**Arguments**

NORMal colors traces according to their channel.

TEMPerature colors all traces using a multicolored palette, where “intensity” is represented by hue, blue for least frequently hit, red for most frequently hit. All traces share this palette. This is the default color palette.

SPECtral colors all traces using a multicolored palette, where “intensity” is represented by hue, red for least frequently hit, blue for most frequently hit. All traces share this palette.

INVErted Inverts the normal display hues and lightness levels based on sample intensity. The areas of lowest sample density appear the brightest, while the areas with the highest sample density appear the darkest.
Examples
ACQUIRE:FASTACQ:PALETTE TEMPerature sets the acquisition fast acquisition palette to Temperature.

ACQUIRE:FASTACq:STATE

Sets or queries the state of fast acquisition mode.

Group Acquisition
Syntax ACQUIRE:FASTACq:STATE {<NR1>|OFF|ON}
ACQUIRE:FASTACq:STATE?
Arguments <NR1> = 0 disables FASTAcq; any other value turns this feature on.
OFF disables the FASTAcq feature.
ON enables the FASTAcq feature.
Examples ACQUIRE:FASTACq:STATE ON turns on fast acquisition mode.
ACQUIRE:FASTACq:STATE? might return :ACQUIRE:FASTACq:STATE 0 indicating that fast acquisition mode is off.

ACQUIRE:MAXSamplerate? (Query Only)

This query returns the maximum real-time sample rate.

Group Acquisition
Syntax ACQUIRE:MAXSamplerate?

ACQUIRE:MODe

This command sets or queries the selected acquisition mode of the instrument.
Commands listed in alphabetical order

**Group**: Acquisition

**Syntax**

```
ACQuire:MODe {SAMple|PEAKdetect|HIRes|AVErage|ENVelope}

ACQuire:MODe?
```

**Related Commands**

- ACQuire:NUMAVg
- CURVe

**Arguments**

- **SAMple** specifies that the displayed data point value is the first sampled value that is taken during the acquisition interval. In sample mode, all waveform data has 8 bits of precision. You can request 16 bit data with a CURVe query but the lower-order 8 bits of data will be zero. **SAMple** is the default mode.

- **PEAKdetect** specifies the display of high-low range of the samples taken from a single waveform acquisition. The high-low range is displayed as a vertical column that extends from the highest to the lowest value sampled during the acquisition interval. **PEAKdetect** mode can reveal the presence of aliasing or narrow spikes.

- **HIRes** specifies Hi Res mode where the displayed data point value is the average of all the samples taken during the acquisition interval. This is a form of averaging, where the average comes from a single waveform acquisition. The number of samples taken during the acquisition interval determines the number of data values that compose the average.

- **AVErage** specifies averaging mode, in which the resulting waveform shows an average of **SAMple** data points from several separate waveform acquisitions. The instrument processes the number of waveforms you specify into the acquired waveform, creating a running exponential average of the input signal. The number of waveform acquisitions that go into making up the average waveform is set or queried using the ACQuire:NUMAVg command.

- **ENVelope** specifies envelope mode, where the resulting waveform shows the **PEAKdetect** range of data points from several separate waveform acquisitions. The number of waveform acquisitions that go into making up the envelope waveform is set or queried using the ACQuire:NUMAVg command.

The instrument acquires data after each trigger event using Sample mode; it then determines the pixmap location of each sample point and accumulates it with stored data from previous acquisitions.

A Pix map is a two dimensional array. The value at each point in the array is a counter that reflects the hit intensity. Infinite and noninfinite persist display modes affect how pix maps are accumulated. Zoom, Math, FastAcq, FastFrame, XY, Roll, and Interpolated Time (IT) Sampling Mode are conflicting features to WFMDB acqMode. Turning on one of them generally turns the other one off. Selection of some standard masks (for example, eye masks, which require option MTM) changes the acquisition mode to WFMDB.
**Examples**

ACQUIRE:MODE ENVELOPE sets the acquisition mode to display a waveform that is an envelope of many individual waveform acquisitions.

ACQUIRE:MODE? might return :ACQUIRE:MODE AVERAGE, indicating that the displayed waveform is the average of the specified number of waveform acquisitions.

---

**ACQUIRE:NUMACq? (Query Only)**

This query-only command returns the number of waveform acquisitions that have occurred since the last time acquisitions were stopped.

**Group**

Acquisition

**Syntax**

ACQUIRE:NUMACq?

**Related Commands**

ACQUIRE:STATE

**Examples**

ACQUIRE:NUMACq? might return :ACQUIRE:NUMACq 350, indicating that 350 acquisitions have occurred since executing an ACQUIRE:STATE RUN command.

---

**ACQUIRE:NUMAVg**

This command sets or queries the number of waveform acquisitions that make up an averaged waveform. Ranges from 2 to 10240.

**Group**

Acquisition

**Syntax**

ACQUIRE:NUMAVg <NR1>

ACQUIRE:NUMAVg?

**Related Commands**

ACQUIRE:MODE

**Arguments**

<NR1> is the number of waveform acquisitions to average.

**Examples**

ACQUIRE:NUMAVG 10 specifies that 10 waveform averages will be performed before exponential averaging starts.

ACQUIRE:NUMAVG? might return :ACQUIRE:NUMAVG 75, indicating that there are 75 acquisitions specified for averaging.
ACQuire:NUMFRAMESACQuired? (Query Only)

This query returns the number of FastFrame frames which have been acquired.

Group Horizontal

Syntax ACQuire:NUMFRAMESACQuired?

Examples ACQUIRE:NUMFRAMESACQUIRED? might return
:ACQUIRE:NUMFRAMESACQUIRED 4 indicating 4 frames have been acquired.

ACQuire:SEQuence:CURrent? (Query Only)

In single sequence acquisition mode, this query returns the number of acquisitions or measurements in the sequence completed so far.

Group Acquisition

Syntax ACQuire:SEQuence:CURrent?


ACQuire:SEQuence:MODe

In single sequence acquisition, the single sequence stop after count is based on the number of acquisitions.

Group Acquisition

Syntax ACQuire:SEQuence:MODe NUMACQs

Arguments NUMACQs is the number of acquisitions.

Examples ACQUIRE:SEQUENCE:MODE? might return :ACQUIRE:SEQUENCE:MODE NUMACQ5 indicating the acquisition sequence mode is set to NUMACQ5.
ACQuire:SEQuence:NUMSEQuence

In single sequence acquisition mode, specify the number of acquisitions or measurements that comprise the sequence. The default is 1.

Group  
Acquisition

Syntax  
ACQuire:SEQuence:NUMSEQuence <NR1>  
ACQuire:SEQuence:NUMSEQuence?

Arguments  
<NR1> is the number of acquisitions or measurements that comprise the sequence.

Examples  
ACQUIRE:SEQUENCE:NUMSEQUENCE 2 sets the number of acquisition in a sequence is set to 2.

ACQUIRE:SEQUENCE:NUMSEQUENCE? might return  
ACQUIRE:SEQUENCE:NUMSEQUENCE 1, indicating that the number of acquisition in a sequence is set to 1.

ACQuire:STATE

This command starts or stops acquisitions. When state is set to ON or RUN, a new acquisition will be started. If the last acquisition was a single acquisition sequence, a new single sequence acquisition will be started. If the last acquisition was continuous, a new continuous acquisition will be started.

If RUN is issued in the middle of completing a single sequence acquisition (for example, averaging or enveloping), the acquisition sequence is restarted, and any accumulated data is discarded. Also, the instrument resets the number of acquisitions. If the RUN argument is issued while in continuous mode, a reset occurs and acquired data continues to acquire.

If acquire:stopafter is SEQUENCE, this command leaves the instrument in single sequence, unlike the run/stop button which takes the instrument out of single sequence.

Group  
Acquisition

Syntax  
ACQuire:STATE {<NR1>|OFF|ON|RUN|STOP}  
ACQuire:STATE?

Related Commands  
ACQuire:STOPAfter
Arguments

<NR1> = 0 stops acquisitions, any other value starts acquisitions.
OFF stops acquisitions.
ON starts acquisitions.
RUN starts acquisitions.
STOP stops acquisitions.

Examples

ACQUIRE:STATE RUN starts the acquisition of waveform data and resets the count of the number of acquisitions.

ACQUIRE:STATE? might return :ACQUIRE:STATE 0, indicating that the acquisition is stopped.

ACQuire:STOPAfter

This command sets or queries whether the instrument continually acquires acquisitions or acquires a single sequence. Pressing SINGLE on the front panel button is equivalent to sending these commands: ACQUIRE:STOPAFTER SEQUENCE and ACQUIRE:STATE 1.

Group

Acquisition

Syntax

ACQuire:STOPAfter {RUNSTop|SEQuence}
ACQuire:STOPAfter?

Related Commands

ACQuire:STATE

Arguments

RUNSTop specifies that the instrument will continually acquire data, if ACQuire:STATE is turned on.
SEQuence specifies that the next acquisition will be a single-sequence acquisition.

Examples

ACQUIRE:STOPAFTER RUNSTOp sets the instrument to continually acquire data.

ACQUIRE:STOPAFTER? might return :ACQUIRE:STOPAFTER SEQUENCE, indicating that the next acquisition the instrument makes will be of the single-sequence type.

AFG:AMPLitude

Sets (or queries) the AFG amplitude in volts, peak to peak.
Commands listed in alphabetical order

**AFG:AMPLitude**

*Syntax*

AFG:AMPLitude <NR3>

AFG:AMPLitude?

*Arguments*

<NR3> is a floating point number that represents the AFG amplitude, peak to peak, in volts.

*Examples*

AFG:AMPLITUDE 1.0 sets the AFG amplitude to 1.0 volts, peak to peak.

AFG:AMPLITUDE? might return :AFG:AMPLITUDE 3.0000 indicating the amplitude is set to 3.0 Volts.

---

**AFG:ARBitrary:SOUrce**

This command sets or queries the source name for the Arbitrary Waveform. Currently supported sources are either waveform file (.wfm) or text file (.csv).

*Syntax*

AFG:ARBitrary:SOUrce <QString>

AFG:ARBitrary:SOUrce?

*Arguments*

<QString> is the source name.

*Examples*

AFG:ARBITRARY:SOURCE "E:/Waveforms/Square.wfm" sets the source waveform to E:/Waveforms/Square.wfm.

AFG:ARBITRARY:SOURCE? might return "E:/Waveforms/Square.wfm" indicating the source is set to E:/Waveforms/Square.wfm.

---

**AFG:FREQuency**

Sets (or queries) the AFG frequency, in Hz.
### AFG:FREQuency

**Syntax**

```
AFG:FREQuency <NR3>
AFG:FREQuency?
```

**Arguments**

- `<NR3>` is the floating point number that represents the AFG frequency, in Hz.

**Examples**

- `AFG:FREQUENCY 100.0E3` sets the AFG frequency to 100 kHz.
- `AFG:FREQUENCY?` might return `:AFG:FREQUENCY 312.5000E+3` indicating the frequency is set to 312.5 kHz.

### AFG:FUNCtion

Sets (or queries) which AFG function to execute.

**Syntax**

```
AFG:FUNCtion {SINE|SQUare|PULSe|RAMP|NOISe|DC|SINC|GAUsSian|LORENTz|ERISe|EDECAy|HAVERSINe|CARDIac|ARBitrary}
AFG:FUNCtion?
```

**Arguments**

- SINE
- SQUare
- PULSe
- RAMP
- NOISe
- DC – The DC level is controlled by `AFG:OFFSet`.
- SINC (Sin(x)/x)
- GAUsSian
- LORENTz
**Commands listed in alphabetical order**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERISe</td>
<td></td>
</tr>
<tr>
<td>EDECAy</td>
<td></td>
</tr>
<tr>
<td>HAVERSINe</td>
<td></td>
</tr>
<tr>
<td>CARDiac</td>
<td></td>
</tr>
<tr>
<td>ARBITrary</td>
<td></td>
</tr>
</tbody>
</table>

**Examples**

AFG:FUNC LOREN specifies to generate the Lorentz function.

AFG:FUNCTION? might return :AFG:FUNCTION SINE indicating the AFG function is set to sine.

### AFG:HIGHLevel

This command sets (or queries) the high level value of the output waveform, in volts, when using the arbitrary function generator feature.

**Conditions**

Requires option AFG.

**Group**

AFG

**Syntax**

AFG:HIGHLevel <NR3>

AFG:HIGHLevel?

**Arguments**

<NR3> is a floating point number that represents the AFG high level value, in volts.

**Examples**

AFG:HIGHLEVEL 1.0 sets the AFG high level value to 1.0 volts.

AFG:HIGHLEVEL? might return :AFG:HIGHLEVEL 1.5000 indicating the high level is set to 1.5 Volts.

### AFG:LOWLevel

This command sets (or queries) the low level value of the output waveform, in volts, when using the arbitrary function generator feature.

**Conditions**

Requires option AFG.

**Group**

AFG
Syntax
AFG:LOWLeve1 <NR3>
AFG:LOWLeve1?

Arguments
NR3 is the floating point number that represents the AFG low level value, in volts.

Examples
AFG:LOWLEVEL 1.0 sets the AFG low level value to 1.00 volts.
AFG:LOWLEVEL? might return :AFG:LOWLEVEL -1.5000 indicating the low level is set to —1.5 Volts.

AFG:NOISEAdd:PERCent

Sets (or queries) the AFG additive noise level as a percentage. Minimum is 0.0%, maximum is 100.0% and increment is 1.0%.

Conditions
Requires option AFG.

Group
AFG

Syntax
AFG:NOISEAdd:PERCent <NR3>
AFG:NOISEAdd:PERCent?

Arguments
Floating point number that represents the AFG additive noise level, as a percentage.

Examples
AFG:NOISEADD:PERCENT 50 sets the AFG additive noise level to 50 percent.
AFG:NOISEADD:PERCENT? might return :AFG:NOISEADD:PERCENT 0 indicating that no noise is added.

AFG:NOISEAdd:STATE

Sets (or queries) the AFG additive noise state.

Conditions
Requires option AFG.

Group
AFG
Commands listed in alphabetical order

**AFG:NOISEAdd:STATE**

Syntax

```
AFG:NOISEAdd:STATE {0|1|OFF|ON}
AFG:NOISEAdd:STATE?
```

Arguments

1 or ON turns on the AFG additive noise state.
0 or OFF turns it off.

Examples

```
AFG:NOISEAdd:STATE ON turns on the additive noise state.
AFG:NOISEAdd:STATE? might return :AFG:NOISEAdd:STATE 0 indicating
the noise additive state is off.
```

**AFG:OFFSet**

Sets (or queries) the AFG offset value, in volts.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:OFFSet <NR3>
AFG:OFFSet?
```

Arguments

<NR3> is a floating point number that represents the AFG offset, in volts.

Examples

```
AFG:OFFSET 1.0 sets the AFG offset to 1.0 volts.
AFG:OFFSET? might return AFG:OFFSET 0.0E+0 indicating there is no offset.
```

**AFG:OUTPut:LOAd:IMPEDance**

Sets (or queries) the AFG output load impedance.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:OUTPut:LOAd:IMPEDance {FIFty|HIGHZ}
AFG:OUTPut:LOAd:IMPEDance?
```
Arguments  
FIFTy sets the output load impedance to 50 Ohms.
HIGHZ sets the output load impedance to the high-impedance state.

Examples  
AFG:OUTP:LOA:IMPED FIFT sets the AFG output load impedance to 50 Ohms.
AFG:OUTPUT:LOAD:IMPEDANCE? might return AFG:OUTPUT:LOAD:IMPEDANCE HIGHZ indicating the load impedance is set to the high impedance state.

**AFG:OUTPut:STATE**

Sets (or queries) the AFG output state.

**Conditions**  
Requires option AFG.

**Group**  
AFG

**Syntax**  
AFG:OUTPut:STATE {0|1|OFF|ON}
AFG:OUTPut:STATE?

**Arguments**  
1 or ON turns on the AFG output state.
0 or OFF turns it off.

**Examples**  
AFG:OUTPut:STATE ON turns on the AFG output state.
AFG:OUTPut:STATE? might return AFG:OUTPut:STATE 1 indicating the AFG output is on.

**AFG:PERIod**

Sets (or queries) the period of the AFG waveform, in seconds.

**Conditions**  
Requires option AFG.

**Group**  
AFG

**Syntax**  
AFG:PERIod <NR3>
AFG:PERIod?
Arguments  
NR3 is the floating point number that represents the AFG period value, in seconds.

Returns  
The query response is returned in high precision NR3 format (up to 12 digits with more than 4 trailing 0 digits after the decimal point is omitted).

Examples  
AFG:PERIOD 1 sets the AFG period value to 1 second.
AFG:PERIOD? might return :AFG:PERIOD 3.2000E-6 indicating the AFG period is set to 3.2 \( \mu \)s.

AFG:PULse:WIDth  
Sets (or queries) the AFG pulse width, in seconds.

Conditions  
Requires option AFG.

Group  
AFG

Syntax  
AFG:PULse:WIDth <NR3>
AFG:PULse:WIDth?

Arguments  
NR3 is the floating point number that represents the pulse width, in seconds.

Examples  
AFG:PULSE:WIDTH 100.0E-6 sets the AFG pulse width to 100 microseconds.
AFG:PULSE:WIDTH? might return :AFG:PULSE:WIDTH 1.0000E-6 indicating the pulse width is set to 1 \( \mu \)s.

AFG:RAMP:SYMmetry  
Sets (or queries) the AFG ramp symmetry in percent. Minimum is 0.0%, maximum is 100.0% and increment is 0.10%.

Conditions  
Requires option AFG.

Group  
AFG

Syntax  
AFG:RAMP:SYMmetry <NR3>
AFG:RAMP:SYMmetry?
Commands listed in alphabetical order

**Arguments**
Floating point number that represents the AFG ramp symmetry, as a percentage.

**Examples**
```
AFG:RAMP:SYMMETRY 50.0 sets the AFG ramp symmetry to 50 percent.
AFG:RAMP:SYMMETRY? might return :AFG:RAMP:SYMMETRY 50.0000 indicating the symmetry is set to 50%.
```

**AFG:SQUare:DUTy**
Sets (or queries) the AFG duty cycle in percent. The minimum is 10.0%, maximum is 90.0% and increment is 0.10%.

**Conditions**
Requires option AFG.

**Group**
AFG

**Syntax**
```
AFG:SQUare:DUTy <NR3>
AFG:SQUare:DUTy?
```

**Arguments**
Floating point number that represents the AFG duty cycle, as a percentage.

**Examples**
```
AFG:SQUARE:DUTY 50.0 sets the AFG duty cycle to 50 percent.
AFG:SQUARE:DUTY? might return :AFG:SQUARE:DUTY 50.0000 indicating the duty cycle is set to 50%.
```

**ALIas**
This command sets or queries the state of alias functionality, and it is identical to the ALIAS:STATE command.

**Group**
Alias

**Syntax**
```
ALIas {OFF|ON|<NR1>}
ALIas?
```

**Related Commands**
ALIas:STATE
Arguments

OFF turns Alias expansion off.

ON turns Alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

<NR1> = 0 disables Alias mode; any other value enables Alias mode.

Examples

ALIAS ON turns the alias feature on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

ALIAS? might return :ALIAS:STATE 1, indicating that the alias feature is on.

ALIas:CATalog? (Query Only)

This query-only command returns a list of the currently defined alias labels, separated by commas. If no aliases are defined, the query returns the string "".

Group Alias

Syntax ALIas:CATalog?

Examples ALIAS:CATALOG? might return the string :ALIAS:CATALOG "SETUP1", "TESTMENU1", "DEFAULT" showing that there are three aliases named SETUP1, TESTMENU1, and DEFAULT.

ALIas:DEFine

This command assigns a sequence of program messages to an alias label. These messages are then substituted for the alias whenever it is received as a command or query, provided that ALIas:STATE is turned on. The query form of this command returns the definitions of a selected alias.

NOTE. Attempting to give two aliases the same name causes an error. To give a new alias the name of an existing alias, the existing alias must first be deleted.

Group Alias

Syntax ALIas:DEFine <QString><,>{<QString>|<Block>}

ALIas:DEFine?

Related Commands ALIas:STATE
Arguments

The first `<QString>` is the alias label.

This label cannot be a command name. Labels must start with a letter and can contain only letters, numbers, and underscores; other characters are not allowed. The label must be less than or equal to 12 characters.

The second `<QString>` or `<Block>` is a complete sequence of program messages.

The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 256 characters.

Examples

```
ALIAS:DEFINE "ST1",":RECALL:SETUP 'C:/mySetup.set';:AUTOSET EXECUTE"
```
defines an alias named “ST1” that recalls a previously saved setup and performs an autoset.

```
ALIAS:DEFINE? "ST1" returns :ALIAS:DEFINE "ST1",#247
:RECALL:SETUP 'C:/mySetup.set';:AUTOSET EXECUTE.
```

**ALIas:DELEte (No Query Form)**

This command removes a specified alias and is identical to ALIas:DELEte:NAMe. An error message is generated if the named alias does not exist.

**Group**

Alias

**Syntax**

```
ALIAS:DELETE <QString>
```

**Related Commands**

*ESR?

```
ALIAS:DELETE:ALL
```

**Arguments**

`<QString>` is the name of the alias to be removed. Using ALIAS:DELETE without specifying an alias causes an execution error. `<QString>` must be a previously defined value.

**Examples**

```
ALIAS:DELETE "SETUP1" deletes the alias named SETUP1.
```

**ALIas:DELEte:ALL (No Query Form)**

This command deletes all existing aliases.
<table>
<thead>
<tr>
<th>Group</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>ALIas:DELEte:ALL</td>
</tr>
</tbody>
</table>

**Related Commands**

- ALIas:DELEte
- ALIas:DELEte:NAMe

**Examples**

ALIAS:DELETE:ALL deletes all existing aliases.

**ALIas:DELEte:NAMe (No Query Form)**

This command removes a specified alias. An error message is generated if the named alias does not exist. This command is identical to ALIas:DELEte.

<table>
<thead>
<tr>
<th>Group</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>ALIas:DELEte:NAMe &lt;QString&gt;</td>
</tr>
</tbody>
</table>

**Arguments**

<QString> is the name of the alias to remove. Using ALIas:DELEte:NAMe without specifying an alias causes an exception error. <QString> must be a previously defined alias.

**Examples**

ALIAS:DELETE:NAMe "STARTUP" deletes the alias named STARTUP.

**ALIas:STATE**

This command turns aliases on or off and is identical to the ALIas command. The query form of this command returns the state of the aliases.

<table>
<thead>
<tr>
<th>Group</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>ALIas:STATE {&lt;NR1&gt;</td>
</tr>
</tbody>
</table>

**Arguments**

<NR1> = 0 turns off aliases; any other value turns on aliases.

OFF turns alias expansion off.
ON turns alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

Examples

ALIAS:STATE OFF turns off the alias feature.

ALIAS:STATE? might return :ALIAS:STATE ON, indicating that alias expansion is currently turned on.

**ALLEv? (Query Only)**

This query-only command prompts the instrument to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue. Use the *ESR? query to enable the events to be returned. This command is similar to repeatedly sending *EVMsg? queries to the instrument.

Group Status and Error

Syntax ALLEV?

Related Commands *ESR?

EVMsg?

Examples ALLEV? might return :ALLEV 2225,"Measurement error, No waveform to measure; "420,"Query UTERMINATED;".

**AUTOSAVEPITIMEOUT**

This command sets or queries the idle time from the programmable interface before auto-save occurs.

Group Miscellaneous

Syntax AUTOSAVEPITIMEOUT <NR1>

Arguments <NR1>
Examples

AUTOSAVEPITIMEOUT 100000 sets the timeout to 100 seconds.

AUTOSAVEPITIMEOUT? might return :AUTOSAVEPITIMEOUT 180000 indicating the time out value is 180 seconds.

**AUTOSAVEUITIMEOUT**

This command sets or queries the idle time from the user interface before auto-save occurs.

**Group**

Miscellaneous

**Syntax**

AUTOSAVEUITIMEOUT <NR1>

AUTOSAVEUITIMEOUT?

**Arguments**

<NR1>

**Examples**

AUTOSAVEUITIMEOUT 4000 sets the timeout to 4 seconds.

AUTOSAVEUITIMEOUT? might return :AUTOSAVEUITIMEOUT 3000 indicating the time out value is 3 seconds.

**AUTOSet (No Query Form)**

This command (no query format) sets the vertical, horizontal, and trigger controls of the instrument to automatically acquire and display the selected waveform.

**Group**

Miscellaneous

**Syntax**

AUTOSet EXECute

**Arguments**

EXECute autosets the displayed waveform; this is equivalent to pressing the front panel Autoset button.

**Examples**

AUTOSEt EXECute autosets the displayed waveform.
AUTOSet:ACQuisition:ENAble

This command sets or queries the Autoset acquisition setting adjustment. Settings affected may include, but not be limited to, acquisition mode, and FastAcq mode.

Group  Miscellaneous

Syntax  AUTOSet:ACQuisition:ENAble {ON|1|OFF|0}
        AUTOSet:ACQuisition:ENAble?

Related Commands  AUTOSet HOrizontal:ENAble
                   AUTOSet TRIGger:ENAble
                   AUTOSet VERTical:ENAble

Arguments  ON or 1 enables Autoset to change acquisition settings.
            OFF or 0 disables Autoset from changing acquisition settings.

Examples  AUTOSet:ACQuisition:ENAble OFF turns off Autoset's adjustment of the acquisition settings.
          AUTOSet:ACQuisition:ENAble? might return 0, indicating Autoset's adjustment of the acquisition settings is turned off.

AUTOSet:ENAble

This command sets or queries the Autoset enable/disable feature. This is useful for classroom purposes where the instructor wants the students to achieve the desired instrument settings without the benefit of the Autoset feature.

Group  Miscellaneous

Syntax  AUTOSet:ENAble {ON|1|OFF|0}
        AUTOSet:ENAble?

Related Commands  AUTOSet ACQuisition:ENAble
                   AUTOSet HOrizontal:ENAble
                   AUTOSet TRIGger:ENAble
                   AUTOSet VERTical:ENAble
Arguments
ON or 1 enables Autoset.
OFF or 0 disables Autoset.

Examples
AUTOSet:ENAble ON turns on the Autoset feature.
AUTOSet:ENAble? might return 1, indicating that Autoset is enabled.

AUTOSet:HORizontal:ENAble
This command sets or queries Autoset’s adjustment of horizontal settings. Settings
affected may include, but not be limited to, horizontal scale, horizontal position,
and horizontal delay mode.

Group
Miscellaneous

Syntax
AUTOSet:HORizontal:ENAble {ON|1|OFF|0}
AUTOSet:HORizontal:ENAble?

Related Commands
AUTOSet:ACQuisition:ENAble
AUTOSet:TRIGger:ENAble
AUTOSet:VERTical:ENAble

Arguments
ON or 1 enables Autoset to change horizontal settings.
OFF or 0 disables Autoset from changing horizontal settings.

Examples
AUTOSet:HORizontal:ENAble OFF turns off Autoset’s adjustment of the
horizontal settings.
AUTOSet:HORizontal:ENAble? might return 0, indicating Autoset’s
adjustment of the horizontal settings is turned off.

AUTOSet:TRIGger:ENAble
This command sets or queries Autoset’s adjustment of trigger settings. Settings
affected may include, but not be limited to, trigger level, trigger source, and
trigger coupling.

Group
Miscellaneous
Syntax

AUTOSet:TRIGger:ENAble {ON|1|OFF|0}
AUTOSet:TRIGger:ENAble?

Related Commands

AUTOSet:ACQuisition:ENAble
AUTOSet:HORizontal:ENAble
AUTOSet:VERTical:ENAble

Arguments

ON or 1 enables Autoset to change trigger settings.
OFF or 0 disables Autoset from changing trigger settings.

Examples

AUTOSet:TRIGger:ENAble OFF turns off Autoset's adjustment of the trigger settings.
AUTOSet:TRIGger:ENAble? might return 0, indicating Autoset's adjustment of the trigger settings is turned off.

AUTOSet:VERTical:ENAble

This command sets or queries Autoset's adjustment of vertical settings. Settings affected may include, but not be limited to, vertical scale, vertical position, and vertical offset.

Group

Miscellaneous

Syntax

AUTOSet:VERTical:ENAble {ON|1|OFF|0}
AUTOSet:VERTical:ENAble?

Related Commands

AUTOSet:ACQuisition:ENAble
AUTOSet:HORizontal:ENAble
AUTOSet:TRIGger:ENAble

Arguments

ON or 1 enables Autoset to change vertical settings.
OFF or 0 disables Autoset from changing vertical settings.

Examples

AUTOSet:VERTical:ENAble OFF turns off Autoset's adjustment of the vertical settings.
Commands listed in alphabetical order

**AUTOSet:VERTical:ENAble?** might return 0, indicating Autoset's adjustment of the vertical settings is turned off.

**AUTOSet:VERTical:OPTIMize**

This command sets or queries which vertical settings Autoset will optimize when the display mode is set to Overlay mode (all waveforms are in one common graticule in the Waveform View).

**Group**  
Miscellaneous

**Syntax**  
`AUTOSet:VERTical:OPTIMize {RESOlution|VISIBility}`  
`AUTOSet:VERTical:OPTIMize?`

**Related Commands**  
`DISplay:WA VEView<x>:VIEWStyle`

**Arguments**  
**RESolution** uses as much of the ADC's (Analog to Digital Converter) range as possible to provide the best vertical resolution and measurement accuracy, but waveforms will overlap each other.

**VISIBility** vertically scales and positions waveforms so they are visually separated from each other at the expense of vertical resolution and measurement accuracy.

**Examples**  
`AUTOSet:VERTical:OPTIMize VISIBility` will vertically scale and position all active waveforms so that they are visually separated from each other when autoset is executed.

`AUTOSet:VERTical:OPTIMize?` might return `AUTOSet:VERTical:OPTIMize RESOLUTION`, which indicates that, when executed, autoset will vertically scale/position the active waveforms to maximize the ADC range while in Overlay display mode.

**AUXout:EDGE**

This command sets or queries the direction in which the Auxiliary Output signal will transition when a trigger occurs.

**Group**  
Miscellaneous
AUXout:EDGE {RISing|FALling}
AUXout:EDGE?

**Arguments**

RISing sets the polarity to the rising edge.
FALling sets the polarity to the falling edge.

**Examples**

AUXOUT:EDGE RISING sets the polarity to rising edge.
AUXOUT:EDGE? might return :AUXOUT:EDGE FALLING, indicating that the polarity is set to falling edge.

**AUXout:SOUrce**

This command sets or queries the source at the Auxiliary Output BNC connection.

**Syntax**

AUXout:SOUrce {ATRIGger|REFOUT|AFG}
AUXout:SOUrce?

**Arguments**

ATRIGger sets the source at the BNC connector to the main trigger.
REFOUT sets the source at the BNC connector to the reference output.
AFG sets the source at the BNC connector to the AFG output.

**Examples**

AUXOUT:SOURCe? might return :AUXOUT:SOURCe ATRIGGER, indicating that the source at the BNC connector is set to the A trigger.

**BUS:ADDNew (No Query Form)**

This command adds the specified bus. This command creates/adds the bus but does not display it (turn it on). In order to enable bus decoding and see the bus display on screen, send the DISplay:WA VEView<x>:BUS:B<x>:STATE ON command.

**Group**

Bus

**Syntax**

BUS:ADDNew <QString>
Related Commands  

BUS:B<x>:TYPE

DISplay WAVEView<x>:BUS B<x>:STATE

Arguments  

<String> is a quoted string of the form "B<NR1>" where NR1 is \( \geq 1 \).

Examples  

BUS:ADDNEW "B5";:display:waveview1:bus:b5:state on creates bus 3 with the default type of Parallel, and then turns it on.

BUS:B<x>:ARINC429A:BITRate

This command sets or queries the ARINC429 bit rate for the specified bus. The bus number is specified by \( x \). If you select Custom, use BUS:B<x>:ARINC429A:BITRate:CUSTom to set the bit rate.

Conditions  

Requires option SR-AERO.

Group  

Bus

Syntax  

BUS:B<x>:ARINC429A:BITRate {LOW|HI|CUSTom}

BUS:B<x>:ARINC429A:BITRate?

Related Commands  

\( B<x> \) is the number of the bus.

BUS:B<x>:ARINC429A:BITRate:CUSTom

Arguments  

Arguments specify the bit rate.

Examples  

BUS:B1:ARINC429A:BITRATE LOW sets the bit rate to handle low speed signals.


BUS:B<x>:ARINC429A:BITRate:CUSTom

This command sets or queries the ARINC429 custom bit rate for the specified bus. The bus is specified by \( x \).

Conditions  

Requires option SR-AERO.
Commands listed in alphabetical order

**BUS:B<x>:ARINC429A:BITRate:CUSTOM**

**Syntax**

BUS:B<x>:ARINC429A:BITRate:CUSTom <NR1>
BUS:B<x>:ARINC429A:BITRate:CUSTom?

**Related Commands**

BUS:B<x>:ARINC429A:BITRate

**Arguments**

B<x> is the number of the bus.

<NR1> is the ARINC429 custom bit rate for the specified bus.

**Examples**

BUS:B1:ARINC429A:BITRATE:CUSTOM 12500 sets the bit rate to 12,500.

**BUS:B<x>:ARINC429A:DATAFORMAT**

This command sets or queries the format of the DATA field for the specified ARINC429 bus. The bus is specified by x.

**Conditions**

Requires option SR-AERO.

**Group**

Bus

**Syntax**

BUS:B<x>:ARINC429A:DATAFORMAT {DATA|SDIDATA|SDIDATASSM}
BUS:B<x>:ARINC429A:DATAFORMAT?

**Arguments**

B<x> is the number of the bus.

DATA specifies a DATA field width of 19 bits (covering bits 11 through 29 of the 32 bit packet)

SDIDATA specifies a DATA field width of 21 bits (covering bits 9 through 29 of the 32 bit packet)

SDIDATASSM specifies a DATA field width of 23 bits (covering bits 9 through 31 of the 32 bit packet)

**Examples**

BUS:B1:ARINC429A:DATAFORMAT SDIDATASSM sets Bus 1 ARINC429 DATA field width to 23 bits.
Commands listed in alphabetical order

**BUS:B<x>:ARINC429A:DATAFORMAT?**

This command returns the DATA field width of an ARINC429 bus specified by x. The maximum return width is 19 bits. For example:

```
```

**BUS:B<x>:ARINC429A:POLARITY**

This command sets or queries the source polarity for the specified ARINC429 bus. The bus is specified by x.

**Conditions**

Requires option SR-AERO.

**Group**

Bus

**Syntax**

```plaintext
BUS:B<x>:ARINC429A:POLARITY {NORMal|INVERTed}
BUS:B<x>:ARINC429A:POLARITY?
```

**Arguments**

- **B<x>** is the number of the bus.
- **NORMal** specifies normal polarity.
- **INVERTed** specifies inverted polarity.

**Examples**

```

:BUS:B2:ARINC429A:POLARITY NORMAL, indicating that the Bus 2 ARINC429 polarity is set to normal.
```

**BUS:B<x>:ARINC429A:SOUrce**

This command sets or queries the source for the specified ARINC429 bus. The bus is specified by x.

**Conditions**

Requires option SR-AERO.

**Group**

Bus

**Syntax**

```plaintext
BUS:B<x>:ARINC429A:SOUrce {CH<X>|Math<x>|REF<x>}
BUS:B<x>:ARINC429A:SOUrce?
```

2-134 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source waveform for the ARINC429 bus.

Math<x> specifies a math waveform as the source waveform for the ARINC429 bus.

REF<x> specifies a reference waveform as the source waveform for the ARINC429 bus.

Examples

BUS:B1:ARINC429A:SOURCE CH1 sets channel 1 as the source for the ARINC429 bus.


**BUS:B<x>:ARINC429A:THRESHold**

This command sets or queries the ARINC429 upper threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-AERO.

Group

Bus

Syntax

BUS:B<x>:ARINC429A:THRESHold <NR3>

BUS:B<x>:ARINC429A:THRESHold?

Related Commands

BUS:B<x>:ARINC429A:SOUrce

Arguments

B<x> is the number of the bus.

<NR3> is the ARINC429 lower threshold for the specified bus.

Examples

BUS:B3:ARINC429A:THRESHOLD -200e−3 sets the Bus 3 ARINC429 upper threshold to -200 mV.

Commands listed in alphabetical order

**BUS:B<x>:AUDio:BITDelay**

This command sets or queries the number of delay bits for the specified AUDIO bus. The bus is specified by x.

**NOTE.** This command only applies to the TDM Audio type.

**Conditions**
Requires option SR-AUDIO.

**Group**
Bus

**Syntax**
BUS:B<x>:AUDio:BITDelay <NR1>
BUS:B<x>:AUDio:BITDelay?

**Arguments**
B<x> is the number of the bus.
<NR1> specifies the number of delay bits.

**Examples**
BUS:B1:AUDIO:BITDELAY 2 sets the bit delay to 2.

**BUS:B<x>:AUDio:BITOrder**

Specifies the bit order for the specified AUDIO bus. The bus is specified by x.

**Conditions**
Requires option SR-AUDIO.

**Group**
Bus

**Syntax**
BUS:B<x>:AUDio:BITOrder {MSB|LSB}
BUS:B<x>:AUDio:BITOrder?

**Arguments**
B<x> is the number of the bus.
MSB specifies that the most significant bit will be expected first in the order.
LSB specifies that the least significant bit will be expected first in the order.
Examples

**BUS:B1:AUDIO:BITORDER** LSB sets the bit order to LSB.
**BUS:B1:AUDIO:BITORDER?** might return :BUS:B1:AUDIO:BITORDER MSB indicating that the MSB is first in the bit order.

**BUS:B<x>:AUDio:CLOCK:POLarity**

This command sets or queries the clock source polarity for the specified AUDIO bus. The bus is specified by x.

**Conditions**
Requires option SR-AUDIO.

**Group**
Bus

**Syntax**
**BUS:B<x>:AUDio:CLOCK:POLarity** {FALL|RISE}
**BUS:B<x>:AUDio:CLOCK:POLarity?**

**Arguments**

B<x> is the number of the bus.
FALL sets falling edge as the clock polarity.
RISE sets rising edge as the clock polarity.

**Examples**

**BUS:B1:AUDIO:CLOCK:POLARITY** FALL sets the clock polarity to Fall.

**BUS:B<x>:AUDio:CLOCK:SOUrce**

This command sets or queries the clock source waveform for the specified AUDIO bus. The bus is specified by x.

**Conditions**
Requires option SR-AUDIO.

**Group**
Bus

**Syntax**
**BUS:B<x>:AUDio:CLOCK:SOUrce**
{CH<x>|CH<x>_D<x>|Math<x>|REF<x>|REF<x>_D<x>}
**BUS:B<x>:AUDio:CLOCK:SOUrce?**
Arguments

- **B<x>** is the number of the bus.
- **CH<x>** specifies an analog channel as the clock source waveform for the audio bus.
- **CH<x>_D<x>** specifies a digital channel as the clock source waveform for the specified audio bus.
- **Math<x>** specifies a math waveform as the clock source waveform for the audio bus.
- **REF<x>** specifies a reference waveform as the clock source waveform for the audio bus.
- **REF<x>_D<x>** specifies a digital reference waveform as the clock source waveform for the specified audio bus.

Examples

```
BUS:B1:AUDIO:CLOCK:SOURCE CH1_D1
```
sets D1 of channel 1 as the clock source for the audio bus.

```
:BUS:B1:AUDIO:CLOCK:SOURCE CH1 indicating that the clock source is set to channel 1.
```

**BUS:B<x>:AUDiO:CLOCk:THReshold**

This command sets or queries the audio clock source threshold for the specified bus. The bus is specified by x.

**Conditions**

Requires option SR-AUDIO.

**Group**

Bus

**Syntax**

```
BUS:B<x>:AUDiO:CLOCk:THReshold <NR3>
BUS:B<x>:AUDiO:CLOCk:THReshold?
```

**Related Commands**

**BUS:B<x>:AUDiO:CLOCk:SOUrce**

**Arguments**

- **B<x>** is the number of the bus.
- **<NR3>** is the audio clock source threshold for the specified bus.

**Examples**

```
BUS:B3:AUDIO:CLOCK:THRESHOLD 500.0E-3 sets the Bus 3 audio clock source threshold to 500.0 mV.
```
BUS:B2:AUDIO:CLOCK:THRESHOLD? might return:
:BUS:B2:AUDIO:CLOCK:THRESHOLD 1.0 indicates the Bus 2 audio clock source threshold is set to 1.0 V.

BUS:B<x>:AUDio:DATa:POLarity

This command sets or queries the audio data source polarity for the specified audio bus. The bus is specified by x.

Conditions
Requires option SR-AUDIO.

Group
Bus

Syntax
BUS:B<x>:AUDio:DATa:POLarity {HIGH|LOW}
BUS:B<x>:AUDio:DATa:POLarity?

Arguments
B<x> is the number of the bus.

HIGH specifies positive data polarity for the audio bus.
LOW specifies negative data polarity for the audio bus.

Examples
BUS:B1:AUDIO:DATA:POLARITY LOW sets the data polarity to LOW.
BUS:B1:AUDIO:DATA:POLARITY? might return:
:BUS:B1:AUDIO:DATA:POLARITY HIGH indicating that the data polarity is set to HIGH.

BUS:B<x>:AUDio:DATa:SIZe

This command sets or queries the number of bits per channel for the specified audio bus. The bus is specified by x.

NOTE. This command only applies to the TDM Audio type.

Conditions
Requires option SR-AUDIO.

Group
Bus
Syntax

BUS:B<x>:AUDio:DATa:SIZe <NR1>
BUS:B<x>:AUDio:DATa:SIZe?

Arguments

B<x> is the number of the bus.

NR1 specifies the number of bits per word.

Examples

BUS:B1:AUDIO:DATA:SIZE 8 sets the number of bits per word to 8.


**BUS:B<x>:AUDio:DATa:SOUrce**

This command sets or queries the audio data source for the specified audio bus. The bus is specified by x.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

BUS:B<x>:AUDio:DATa:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:AUDio:DATa:SOUrce?

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the data source waveform for the audio bus.

CH<x>_D<x> specifies a digital channel as the data source waveform for the audio bus.

MATH<x> specifies an math waveform as the data source waveform for the audio bus.

REF<x> specifies an reference waveform as the data source waveform for the audio bus.

REF<x>_D<x> specifies a digital reference waveform as the data source waveform for the specified audio bus.

Examples

BUS:B1:AUDIO:DATA:SOURCE CH1_D1 sets the data source to D1 of FlexChannel 1.
Commands listed in alphabetical order

BUS:B<x>:AUDio:DATa:THReshold

This command sets or queries the audio data source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-AUDIO.

Group Bus

Syntax BUS:B<x>:AUDio:DATa:THReshold <NR3>
BUS:B<x>:AUDio:DATa:THReshold?

Related Commands BUS:B<x>:AUDio:DATa:SOURce

Arguments B<x> is the number of the bus.
<NR3> is the audio data source threshold for the specified bus.

Examples BUS:B3:AUDIO:DATA:THRESHOLD 1.5 sets the Bus 3 audio clock source threshold to 500.0 mV.

BUS:B<x>:AUDio:DATa:WORDSize

This command sets or queries the audio bits per word for the specified bus. The bus is specified by x.

Group Bus

Syntax BUS:B<x>:AUDio:DATa:WORDSize <NR1>
BUS:B<x>:AUDio:DATa:WORDSize?

Examples
Arguments

B<x> is the number of the bus.

<NR1> is the audio bits per word for the specified bus.

Examples

BUS:B1:AUDIO:DATA:WORDSize 24 sets the audio bits per word to 24 bits.

:BUS:B1:AUDIO:DATA:WORDSIZE 24 indicating the bits per word is 24.

**BUS:B<x>:AUDio:FRAME:CLOCKBITSPERCHANNEL**

This command sets or queries the audio bits of sync width for the specified bus. The bus is specified by x.

**NOTE. This command only applies to the TDM Audio type.**

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

BUS:B<x>:AUDio:FRAME:CLOCKBITSPERCHANNEL <NR1>

BUS:B<x>:AUDio:FRAME:CLOCKBITSPERCHANNEL?

Arguments

B<x> is the number of the bus.

<NR1> is the audio bits of sync width for the specified bus.

Examples

BUS:B1:AUDio:FRAME:CLOCKBITSPERCHANNEL 32 sets the number of bits to 32.

:BUS:B1:AUDIO:FRAME:CLOCKBITSPERCHANNEL 24 indicating there are 24 bits of sync width for the bus.

**BUS:B<x>:AUDio:FRAME:SIZe**

This command sets or queries the number of audio channels in each frame for the specified AUDIO bus. The bus is specified by x.

**NOTE. This command only applies to the TDM Audio type.**
Commands listed in alphabetical order

**Conditions**
Requires option SR-AUDIO.

**Group**
Bus

**Syntax**

```
BUS:B<x>:AUDio:FRAME:SIZe <NR1>
BUS:B<x>:AUDio:FRAME:SIZe?
```

**Arguments**

- `B<x>` is the number of the bus.
- `<NR1>` specifies the number of channels in each frame.

**Examples**

- `BUS:B1:AUDIO:FRAME:SIZE 2` sets the frame size to 2.

**BUS:B<x>:AUDio:TYPe**

This command sets or queries the audio format (type) for the specified audio bus. The bus is specified by `x`.

**Conditions**
Requires option SR-AUDIO.

**Group**
Bus

**Syntax**

```
BUS:B<x>:AUDio:TYPe {I2S|LJ|RJ|TDM}
BUS:B<x>:AUDio:TYPe?
```

**Arguments**

- `B<x>` is the number of the bus.
- `I2S` specifies the I2S audio format.
- `LJ` specifies the left-justified audio format.
- `RJ` specifies the right-justified audio format.
- `TDM` specifies the time-division multiplexing audio format.

**Examples**

- `BUS:B1:AUDIO:TYPe RJ` sets right-justified as the audio format.
- `BUS:B1:AUDIO:TYPe?` might return `BUS:B1:AUDIO:TYPe I2S` indicating that the audio format is set to `I2S`. 
BUS:B<x>:AUDio:WORDSel:POLarity

This command sets or queries the word select source polarity for the specified audio bus. The bus is specified by x.

**Conditions**
Requires option SR-AUDIO.

**Group**
Bus

**Syntax**
BUS:B<x>:AUDio:WORDSel:POLarity {NORMal|INVERTed}
BUS:B<x>:AUDio:WORDSel:POLarity?

**Arguments**
B<x> is the number of the bus.
NORMal specifies positive polarity.
INVERTed specifies negative polarity.

**Examples**
BUS:B1:AUDIO:WORDSEL:POLARITY NORMal sets normal as the word select polarity.

---

BUS:B<x>:AUDio:WORDSel:SOUrce

This command sets or queries the audio word select source waveform for the specified audio bus. The bus is specified by x.

**Conditions**
Requires option SR-AUDIO.

**Group**
Bus

**Syntax**
BUS:B<x>:AUDio:WORDSel:SOUrce {CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:AUDio:WORDSel:SOUrce?

**Arguments**
B<x> is the number of the bus.
CH<x> specifies an analog channel as the word select source waveform.
CH<x>_D<x> specifies a digital channel as the word select source waveform.
MATH<x> specifies an math waveform as the word select source waveform.
REF<x> specifies an reference waveform as the word select source waveform.
REF<x>_D<x> specifies a digital reference waveform as the word select source waveform for the specified audio bus.

Examples

BUS:B1:AUDIO:WORDSEL:SOURCE CH1 sets CH1 as the word select source.
BUS:B1:AUDIO:WORDSEL:SOURCE CH2_D3 sets D3 of FlexChannel 2 as the word select source.

**BUS:B<x>:AUDio:WORDSel:THReshold**

This command sets or queries the audio word select source threshold for the specified bus. The bus is specified by x.

**Conditions**
Requires option SR-AUDIO.

**Group**
Bus

**Syntax**
BUS:B<x>:AUDio:WORDSel:THReshold <NR3>
BUS:B<x>:AUDio:WORDSel:THReshold?

**Arguments**
B<x> is the number of the bus.
<NR3> is the audio word select source threshold for the specified bus.

**Examples**
BUS:B4:AUDIO:WORDSEL:THRESHOLD 300.0E-3 sets the Bus 4 audio word select source threshold to 300 mV.
BUS:B1:AUDIO:WORDSEL:THRESHOLD? might return :
BUS:B1:AUDIO:WORDSEL:THRESHOLD 1.25 indicates the Bus 1 audio word select source threshold is set to 1.25 V.
BUS:B<x>:CAN:BITRate

This command sets or queries the CAN bit rate. The bus number is specified by x. If you select Custom, use BUS:B<x>:CAN:BITRate:VALUE to set the bit rate.

Conditions
Requires option SR-AUTO.

Group
Bus

Syntax
BUS:B<x>:CAN:BITRate
{RATE10K|RATE100K|RATE1M|RATE125K|RATE153K|
RATE20K|RATE25K|RATE250K|RATE31K|RATE33K|RATE400K|RATE50K|
RATE500K|RATE62K|RATE68K|RATE800K|RATE83K|RATE92K|CUSTOM}
BUS:B<x>:CAN:BITRate?

Related Commands
BUS:B<x>:CAN:BITRate:VALUE

Arguments
B<x> is the number of the bus.
Arguments specify the bit rate.

Examples
BUS:B1:CAN:BITRate RATE1M sets the CAN bit rate to 1 Mb.

BUS:B<x>:CAN:BITRate:VALUE

This command sets or queries CAN bit rate when Custom is selected by BUS:B<x>:CAN:BITRate. The bus number is specified by x.

Conditions
Requires option SR-AUTO.

Group
Bus

Syntax
BUS:B<x>:CAN:BITRate:VALUE <NR3>
BUS:B<x>:CAN:BITRate:VALUE?

Related Commands
BUS:B<x>:CAN:BITRate
Arguments

B<x> is the number of the bus.

<NR3> specifies the CAN bit rate.

Returns

<NR1> is the bit rate.

Examples

BUS:B1:CAN:BITRate:VALUE 400.0E+3 sets the bit rate to 400 k.


BUS:B<x>:CAN:FD:BITRate

This command sets or queries the increased data phase bit rate used by CAN FD packets on the specified CAN bus. The bus is specified by x. If you select Custom, use BUS:B<x>:CAN:FD:BITRate:CUSTom to set the bit rate.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

BUS:B<x>:CAN:FD:BITRate
{RATE1M|RATE2M|RATE3M|RATE4M|RATE5M|RATE6M|RATE7M|RATE8M|RATE9M|RATE10M|RATE11M|RATE12M|RATE13M|RATE14M|RATE15M|RATE16M|CUSTom}

BUS:B<x>:CAN:FD:BITRate?

Related Commands

BUS:B<x>:CAN:FD:BITRate:CUSTom

Arguments

B<x> is the number of the bus.

Arguments specify the bit rate.

Examples

BUS:B1:CAN:BITRATE RATE1M sets the CAN FD bit rate to 1 Mbps.

**BUS:B<x>:CAN:FD:BITRate:CUSTom**

This command sets or queries the custom bit rate for the increased data phase of CAN FD packets on the specified CAN bus. The bus is specified by x.

- **Conditions**: Requires option SR-AUTO.
- **Group**: Bus
- **Syntax**:
  - `BUS:B<x>:CAN:FD:BITRate:CUSTom <NR1>`
  - `BUS:B<x>:CAN:FD:BITRate:CUSTom?`
- **Arguments**:
  - `<B<x>` is the number of the bus.
  - `<NR1>` is the custom FD bit rate for the specified bus.
- **Examples**:
  - `BUS:B1:CAN:FD:BITRATE:CUSTOM 1000000` sets the bit rate to 1,000,000.

---

**BUS:B<x>:CAN:SAMPLEpoint**

This command sets or queries the sample point for the specified CAN bus. The bus is specified by x.

- **Conditions**: Requires option SR-AUTO.
- **Group**: Bus
- **Syntax**:
  - `BUS:B<x>:CAN:SAMPLEpoint <NR1>`
  - `BUS:B<x>:CAN:SAMPLEpoint?`
- **Arguments**:
  - `<B<x>` is the number of the bus.
  - `<NR1>` is the sample point, in percent, for the specified CAN bus.
Examples

`BUS:B1:CAN:SAMPLEpoint 40` sets the sample point to 40%.

`BUS:B1:CAN:SAMPLEpoint?` might return `:BUS:B1:CAN:SAMPLEPOINT 50` indicating the sample point is at 50%.

**BUS:B<x>:CAN:SIGNal**

This command sets or queries the signal type for the specified CAN bus. The bus is specified by \( x \).

**Conditions**

Requires option SR-AUTO.

**Group**

Bus

**Syntax**

`BUS:B<x>:CAN:SIGNal {DIFFerential|CANH|CANL|RX|TX}`

`BUS:B<x>:CAN:SIGNal?`

**Arguments**

\( B<x> \) is the number of the bus.

Arguments are the CAN bus signal types.

**Examples**

`BUS:B1:CAN:SIGNal CANL` sets the signal type to CANL.

`BUS:B1:CAN:SIGNal?` might return `:BUS:B1:CAN:SIGNAL CANH` indicating the signal type is CANH.

**BUS:B<x>:CAN:SOUrce**

This command sets or queries the CAN source channel for the specified CAN bus. The bus number is specified by \( x \).

**Conditions**

Requires option SR-AUTO.

**Group**

Bus

**Syntax**

`BUS:B<x>:CAN:SOUrce {CH<x> | CH<x>_D<x> | MATH<x> | REF<x> | REF<x>_D<x>}`

`BUS:B<x>:CAN:SOUrce?`
Arguments

B<x> is the number of the bus.

Arguments specify a channel.

Examples

BUS:B1:CAN:SOURCEx MATH6 sets the source channel to MATH6.

BUS:B1:CAN:SOURCEx? might return :BUS:B1:CAN:SOURCEx CH1, indicating that the CAN source channel is CH1.

BUS:B<x>:CAN:STANDard

This command sets or queries which CAN standard specification to analyze the specified CAN bus with. The bus is specified by x.

Conditions

Requires option SR-AERO.

Group

Bus

Syntax

BUS:B<x>:CAN:STANDard {CAN2X|FDISO|FDNONISO}

BUS:B<x>:CAN:STANDard?

Arguments

B<x> is the number of the bus.

CAN2X sets the CAN bus standard to CAN 2.0.

FDISO sets the CAN bus standard to ISO CAN FD (11898-1:2015).

FDNONISO sets the CAN bus standard to non-ISO CAN FD (Bosch 2012).

Examples

BUS:B1:CAN:STANDard FDISO sets the CAN standard to ISO CAN FD.


BUS:B<x>:CAN:THReshold

This command sets or queries the source channel threshold for the specified CAN bus. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus
Syntax

BUS:B<x>:CAN:THReshold <NR3>
BUS:B<x>:CAN:THReshold?

Arguments

B<x> is the number of the bus.

<NR3> is the source channel threshold for the specified CAN bus.

Examples

BUS:B1:CAN:THReshold 5 sets the threshold to 5 V.

BUS:B1:CAN:THReshold? might return :BUS:B1:CAN:THRESHOLD 0.0E+0 indicating the threshold is set to 0.0 V.

BUS:B<x>:DISplay:FORMat

This command sets or queries how the data is represented in the bus form for the specified bus. The bus is specified by x.

Group

Bus

Syntax

BUS:B<x>:DISplay:FORMat
{HEX|BINARY|MIXEDASCII|MIXEDHEX|ASCII|DECIMAL|MIXED}
BUS:B<x>:DISplay:FORMat?

Arguments

B<x> is the number of the bus.

NOTE. Different bus types support only a subset of these arguments.

HEX specifies hexadecimal format.

BINARY specifies binary format.

MIXEDASCII specifies mixed ASCII format.

MIXEDHEX specifies mixed hexadecimal format.

ASCII specifies ASCII format.

DECIMAL specifies decimal format.

MIXED specifies mixed format.

Examples

BUS:B1:DISplay:FORMat BINARY sets the format to binary.

**BUS:B<x>:DISplay:LAYout**

This command sets or queries what to display for the specified bus. The bus is specified by x.

**Group**  
Bus

**Syntax**  
BUS:B<x>:DISPLAY:LAYout {BUS|BUSANDWAVEFORM}  
BUS:B<x>:DISPLAY:LAYout?

**Arguments**  
B<x> is the number of the bus.

BUS specifies displaying the bus form only.

BUSANDWAVEFORM specifies displaying the bus form and the constituent source waveform(s). This argument is not available for some bus types and some bus configurations.

**Examples**  
BUS:B1:DISPLAY:LAYout BUS specifies displaying the bus form only.

BUS:B1:DISPLAY:LAYout? might return :BUS:B1:DISPLAY:LAYOUT BUS indicating the bus will be displayed without displaying its constituent source waveform(s).

**BUS:B<x>:ETHERnet:DATAMINUSTHRESHold**

This command sets or queries the Ethernet D- source threshold for the specified bus. This threshold only applies when the Ethernet signal type is single ended. The bus is specified by x.

**Conditions**  
Requires option SR-ENET.

**Group**  
Bus

**Syntax**  
BUS:B<x>:ETHERnet:DATAMINUSTHRESHold <NR3>  
BUS:B<x>:ETHERnet:DATAMINUSTHRESHold?

**Related Commands**  
BUS:B<x>:ETHERNET:SOURce:DMINus  
BUS:B<x>:ETHERNET:SOURce:DPLUS
Arguments  

\( b \langle x \rangle \) is the number of the bus.

\(<N3>\) is the Ethernet D- source threshold for the specified bus.

Examples  

BUS: B3: ETHERNET: DATAMINUSTHRESHOLD \( 250.0E-3 \) sets the Bus 3 Ethernet DATA Minus source threshold to \( 250.0000 \) mV.


BUS: B2: ETHERNET: DATAMINUSTHRESHOLD 1.0 indicates the Bus 2 Ethernet D- source threshold is set to 1.0 V.

**BUS:B<x>:ETHERnet:DATAPLUSTHRESHold**

This command sets or queries the Ethernet D+ source threshold for the specified bus. This threshold only applies when the Ethernet signal type is single ended. The bus is specified by \( x \).

Conditions  

Requires option SR-ENET.

Group  

Bus

Syntax  

BUS:B<x>:ETHERnet:DATAPLUSTHRESHold <N3>

BUS:B<x>:ETHERnet:DATAPLUSTHRESHold?

Related Commands  

\( b \langle x \rangle \) is the number of the bus.

BUS:B<x>:ETHERNET:SOURce:DMINus

BUS:B<x>:ETHERNET:SOURce:DPLUS

BUS:B<x>:ETHERnet:SIGNALTYpe

Arguments  

\(<N3>\) is the Ethernet D+ source threshold for the specified bus.

Examples  

BUS:B3: ETHERNET:DATAPLUSTHRESHOLD \( 250.0E-3 \) sets the Bus 3 Ethernet D+ source threshold to \( 250.0000 \) mV.


BUS:B2: ETHERNET:DATAPLUSTHRESHOLD 1.0 indicating the Bus 2 Ethernet D+ source threshold is set to 1.0 V.
**BUS:B<x>:ETHERnet:IPVFOUR**

This command sets or queries whether IPv4 packets are available for triggering on Ethernet. The bus is specified by x.

**Conditions**
Requires option SR-ENET.

**Group**
Bus

**Syntax**

\[
\text{BUS:B<x>:ETHERnet:IPVFOUR \{YES|NO\}}
\]

\[
\text{BUS:B<x>:ETHERnet:IPVFOUR?}
\]

**Arguments**

B<x> is the number of the bus.

YES specifies that IPv4 packets are available.

NO specifies that IPv4 packets are not available.

**Examples**

BUS:B1:ETHERNET:IPVFOUR NO will specify that Bus 1 does not have IPv4 packets available.


**BUS:B<x>:ETHERnet:LOWTHRESHold**

This command sets or queries the Ethernet source Low threshold for the specified bus. This threshold only applies when the Ethernet signal type is differential. The bus is specified by x.

**Conditions**
Requires option SR-ENET.

**Group**
Bus

**Syntax**

\[
\text{BUS:B<x>:ETHERnet:LOWTHRESHold \{NR3\}}
\]

\[
\text{BUS:B<x>:ETHERnet:LOWTHRESHold?}
\]

**Related Commands**

BUS:B<x>:ETHERnet:SIGNALTYPE
Arguments
B<x> is the number of the bus.
NR3 is the Ethernet source Low threshold for the specified bus.

Examples
BUS:B1:ETHERnet:LOWTHRESHold -200e-3 sets the threshold to -200 mV.
BUS:B1:ETHERnet:LOWTHRESHold? might return
:BUS:B1:ETHERNET:LOWTHRESHOLD -500.0000E-3 indicating the threshold is set to -500 mV.

BUS:B<x>:ETHERnet:QTAGGING

This command sets or queries whether Q-Tagging packets are available for triggering on Ethernet. The bus is specified by x.

Conditions
Requires option SR-ENET.

Group
Bus

Syntax
BUS:B<x>:ETHERnet:QTAGGING {YES|NO}
BUS:B<x>:ETHERnet:QTAGGING?

Arguments
B<x> is the number of the bus.
YES specifies that Q-Tagging packets are available.
NO specifies that Q-Tagging packets are not available.

Examples
BUS:B1:ETHERnet:QTAGGING YES turns on Q-Tagging.

BUS:B<x>:ETHERnet:SIGNALTYpe

This command sets or queries the Ethernet signal type for the specified bus. The bus is specified by x.

Conditions
Requires option SR-ENET.

Group
Bus
Commands listed in alphabetical order

Syntax

BUS:B<x>:ETHERnet:SIGNALType {SINGLE|DIFF}
BUS:B<x>:ETHERnet:SIGNALType?

Arguments

B<x> is the number of the bus.
SINGLE specifies single-ended signals.
DIFF specifies differential signals.

Examples

BUS:B1:ETHERnet:SIGNALType SINGLE specifies single-ended signals.

BUS:B<x>:ETHERnet:SOURce

This command sets or queries the Ethernet data (SDATA) source for the specified bus. This command controls the source channel when the signal type is differential. The bus number is specified by <x>.

Conditions

Requires option SR-ENET.

Group

Bus

Syntax

BUS:B<x>:ETHERnet:SOURce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ETHERnet:SOURce?

Related Commands

BUS:B <x> ETHERnet THRESHold

Arguments

B<x> is the number of the bus.
CH<x> specifies to use one of the analog channels as the Ethernet data source for differential input.
MATH<x> specifies to use a math waveform as the source for Ethernet data differential input
REF<x> specifies to use one of the reference waveforms as the Ethernet data source for differential input.
Examples

BUS:B1:ETHERNET:SOURce:DMINus CH4 specifies to use the channel 4 waveform as the source for Ethernet data.

BUS:B1:ETHERNET:SOURce:DMINus? might return CH2, indicating that channel 2 is the currently specified source for Ethernet data.

**BUS:B<x>:ETHERNET:SOURce:DMINus**

This command sets or queries the Ethernet D- source for the specified bus. This command specifies the source channel to use when the signal type is single ended. The bus is specified by x.

**Group**

Bus

**Syntax**

BUS:B<x>:ETHERNET:SOURce:DMINus {CH<x>|MATH<x>|REF<x>}

BUS:B<x>:ETHERNET:SOURce:DMINus?

**Arguments**

B<x> is the number of the bus.

CH<x>, MATH<x> or REF<x> set the D- source to the specified signal source.

**Examples**

BUS:B1:ETHERNET:SOURce:DMINus CH1 sets the D- source to channel 1.

BUS:B1:ETHERNET:SOURce:DMINus? might return

:BUS:B1:ETHERNET:SOURce:DMINus CH2 indicating the D- source is set to channel 2.

**BUS:B<x>:ETHERNET:SOURce:DPLUs**

This command sets or queries the Ethernet D+ source for the specified bus. This command specifies the source channel to use when the signal type is single ended. The bus is specified by x.

**Group**

Bus

**Syntax**

BUS:B<x>:ETHERNET:SOURce:DPLUs {CH<x>|MATH<x>|REF<x>}

BUS:B<x>:ETHERNET:SOURce:DPLUs?

**Arguments**

B<x> is the number of the bus.

CH<x>, MATH<x> or REF<x> set the D+ source to the specified signal source.
**Examples**

BUS:B1:ETHERNET:SOURCE:DPLUS Ch5 sets the D+ source to channel 5.


**BUS:B<x>:ETHERnet:THRESHold**

This command sets or queries the Ethernet DATA source High threshold for the specified bus. The bus is specified by x.

**Conditions**

Requires option SR-ENET.

**Group**

Bus

**Syntax**

BUS:B<x>:ETHERnet:THRESHold <NR3>

BUS:B<x>:ETHERnet:THRESHold?

**Related Commands**

BUS:B<x>:ETHERnet:SOURce

**Arguments**

B<x> is the number of the bus.

<NR3> is the Ethernet DATA source High threshold for the specified bus.

**Examples**

BUS:B4:ETHERNET:THRESHOLD 1.0 sets the Bus 4 Ethernet DATA source High threshold to 1.0 V.

BUS:B3:ETHERNET:THRESHOLD? might return

BUS:B3:ETHERNET:THRESHOLD 225.00000E-3 indicates the Bus 3 Ethernet DATA source High threshold is set to 225.0 mV.

**BUS:B<x>:ETHERnet:TYPe**

This command specifies the Ethernet standard speed. The bus number is specified by <x>.

**Conditions**

Requires option SR-ENET.

**Group**

Bus
Syntax

BUS:B<x>:ETHERnet:TYPE \{TENBASET|HUNDREDBASETX\}
BUS:B<x>:ETHERnet:TYPE?

Arguments

B<x> is the number of the bus.

TENBASET specifies the Ethernet speed as 10Base-T.

HUNDREDBASETX specifies the Ethernet speed as 100Base-T.

Examples

BUS:B1:ETHERNET:TYPE HUNDREDBASETX specifies the Ethernet speed as 100Base-T.

BUS:B1:ETHERNET:TYPE? might return TENBASET, indicating that 10Base-T is the currently specified Ethernet speed.

BUS:B<x>:FLEXray:BITRate

This command sets or queries the FlexRay bus bit rate. The bus is specified by x. If you select Custom, use BUS:B<x>:FLEXray:BITRate:CUSTom to set the bit rate.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

BUS:B<x>:FLEXray:BITRate \{CUSTOM|RATE2M|RATE5M|RATE10M\}
BUS:B<x>:FLEXray:BITRate?

Arguments

B<x> is the number of the bus.

Arguments specify the bit rate.

Examples

BUS:B1:FLEXRAY:BITRate RATE2M sets the bit rate to 2 Mb.


BUS:B<x>:FLEXRay:BITRate:CUSTom

This command sets or queries the FlexRay custom bit rate for the specified bus. The bus is specified by x.
Commands listed in alphabetical order

**BUS:B<x>:FLEXray:BITRate:CUSTom**

| Conditions | Requires option SR-AUTO. |
| Group | Bus |
| Syntax | BUS:B<x>:FLEXray:BITRate:CUSTom <NR1>  
BUS:B<x>:FLEXray:BITRate:CUSTom? |
| Arguments | B<x> is the number of the bus.  
<NR1> is the FlexRay custom bit rate for the specified bus. |
| Examples | BUS:B1:FLEXray:BITRate:CUSTom 10000000 sets the bit rate to 10,000,000.  
BUS:B1:FLEXray:BITRate:CUSTom? might return :BUS:B1:FLEXRAY:BITRATE:CUSTOM 10000000 indicating the bit rate is set to 10,000,000. |

**BUS:B<x>:FLEXray:CHannel**

This command sets or queries the FlexRay channel type for the specified bus. The bus number is specified by x.

| Conditions | Requires option SR-AUTO. |
| Group | Bus |
| Syntax | BUS:B<x>:FLEXray:CHannel {A|B}  
BUS:B<x>:FLEXray:CHannel? |
| Arguments | B<x> is the number of the bus.  
A specifies the A channel.  
B specifies the B channel. |
| Examples | BUS:B1:FLEXRAY:CHANNEL B sets the FlexRay channel to B.  
BUS:B<x>:FLEXray:LOWTHRESHold

This command sets or queries the FlexRay data source low threshold for the specified bus. The bus is specified by x.

**Conditions**
Requires option SR-AUTO.

**Group**
Bus

**Syntax**
BUS:B<x>:FLEXray:LOWTHRESHold <NR3>
BUS:B<x>:FLEXray:LOWTHRESHold?

**Arguments**
<NR3> is the FlexRay data source low threshold for the specified bus.
B<x> is the number of the bus.

**Examples**
BUS:B1:FLEXray:LOWTHRESHold 50.0e-3 sets the threshold to 50 mV.
BUS:B1:FLEXray:LOWTHRESHold? might indicate the threshold is set to 0.0 V.

BUS:B<x>:FLEXray:SIGnal

This command sets or queries the FlexRay signal type for the specified bus. The bus number is specified by x.

**Conditions**
Requires option SR-AUTO.

**Group**
Bus

**Syntax**
BUS:B<x>:FLEXray:SIGnal {BDIFFBP|BM|TXRX}
BUS:B<x>:FLEXray:SIGnal?

**Arguments**
B<x> is the number of the bus.
BDIFFBP sets the FlexRay signal type to BDIFFBP.
BM sets the FlexRay signal type to BM.
TXRX sets the FlexRay signal type to TXRX.
Examples

BUS:B1:FLEXRAY:SIGNAL BM sets the FlexRay channel type to BM.
BUS:B1:FLEXRAY:SIGNAL? might return :BUS:B1:FLEXRAY:SIGNAL BDIFFBP, indicating the FlexRay channel type is set to BDIFFBP.

BUS:B<x>:FLEXray:SOURce

This command sets or queries the FlexRay bus data source for the specified bus when the signal type is BDIFFBP or BM. The bus number is specified by x.

Conditions
Requires option SR-AUTO.

Group
Bus

Syntax
BUS:B<x>:FLEXray:SOURce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:FLEXray:SOURce?

Arguments
B<x> is the number of the bus.
Arguments are the available FlexRay sources.

Examples

BUS:B<x>:FLEXray:SOURce:TXRX

This command sets or queries the FlexRay TxRx data source for the specified bus when the signal type is TXRX. The bus is specified by x.

Conditions
Requires option SR-AUTO.

Group
Bus

Syntax
BUS:B<x>:FLEXray:SOURce:TXRX
{CH<x> | CH<x>_D<x> | MATH<x> | REF<x> | REF<x>_D<x>}

Arguments
Arguments are the available sources.
B<x> is the number of the bus.
Examples

**BUS:B1:FLEXray:SOURCE:TXRX CH1** sets the TXRX source the channel 1.

**BUS:B1:FLEXray:SOURCE:TXRX?** might return

:BUS:B1:FLEXRAY:SOURCE:TXRX CH1_D0 indicating the TXRX source is set to CH1_D0.

**BUS:B<x>:FLEXray:THRESHold**

This command sets or queries the FlexRay data source high threshold for the specified bus. The bus is specified by x.

**Conditions**
Requires option SR-AUTO.

**Group**
Bus

**Syntax**

BUS:B<x>:FLEXray:THRESHold <NR3>

BUS:B<x>:FLEXray:THRESHold?

**Arguments**

B<x> is the number of the bus.

<NR3> is the FlexRay data source high threshold for the specified bus.

**Examples**

**BUS:B1:FLEXray:THRESHold 50.0-3** sets the high threshold to 50 mV.

**BUS:B1:FLEXray:THRESHold?** might return **:BUS:B1:FLEXRAY:THRESHOLD 0.0E+0** indicating the high threshold is set to 0.0 V.

**BUS:B<x>:FLEXray:TXRXTHRESHold**

This command sets or queries the FlexRay data source TxRx threshold for the specified bus. The bus is specified by x.

**Conditions**
Requires option 5-SREMBD or SUP5-SREMBD.

**Group**
Bus

**Syntax**

BUS:B<x>:FLEXray:TXRXTHRESHold <NR3>

BUS:B<x>:FLEXray:TXRXTHRESHold?
**Arguments**

B<x> is the number of the bus.

<NR3> is the TxRx threshold.

**Examples**

BUS:B1:FLEXray:TXRXTRESHold 5.0e-3 sets the threshold to 50 mV.

BUS:B1:FLEXray:TXRXTRESHold? might return

:BUS:B1:FLEXRAY:TXRXTRESHOLD 0.0E+0 indicating the threshold is set to 0.0 V.

**BUS:B<x>:I2C:CLOCk:SOUrce**

This command sets or queries the I2C clock (SCLK) source for the specified bus. The bus is specified by x.

**Conditions**

Requires option 5-SREMBD or SUP5-SREMBD.

**Group**

Bus

**Syntax**

BUS:B<x>:I2C:CLOCk:SOUrce {CH<x> | CH<x>_D<x> | MATH<x> | REF<x> | REF<x>_D<x>}

BUS:B<x>:I2C:CLOCk:SOUrce?

**Related Commands**

BUS:B<x>:I2C:DATa:SOUrce

BUS:B<x>:I2C:RWINADDR

**Arguments**

B<x> is the number of the bus.

CH<x> specifies an analog channel to use as the I2C SCLK source.

CH<x>_D<x> specifies a digital channel to use as the I2C SCLK source.

MATH<x> specifies a math waveform to use as the I2C SCLK source.

REF<x> specifies a reference waveform to use as the I2C SCLK source.

REF<x>_D<x> specifies a digital reference waveform as the clock source waveform for the specified I2C bus.

**Examples**

BUS:B1:I2C:CLOCk:SOUrce CH1 sets the I2C SCLK source to CH1.

**BUS:B<x>:I2C:CLOCK:THReshold**

This command sets or queries the I2C Clock (SCLK) source threshold for the specified bus. The bus is specified by x.

**Conditions**
Requires option 5-SREMBD or SUP5-SREMBD.

**Group**
Bus

**Syntax**
BUS:B<x>:I2C:CLOCK:THReshold <NR3>
BUS:B<x>:I2C:CLOCK:THReshold?

**Arguments**
B<x> is the number of the bus.
<NR3> is the I2C Clock (SCLK) source threshold for the specified bus.

**Examples**
BUS:B1:I2C:CLOCK:THReshold 50.0e-3 sets the threshold to 50 mV.
BUS:B1:I2C:CLOCK:THReshold? might return :BUS:B1:I2C:CLOCK:THRESHOLD 0.0E+0 indicating the threshold is set to 0 V.

**BUS:B<x>:I2C:DATa:SOUrce**

This command sets or queries the I2C data (SDA) source for the specified I2C bus. The bus is specified by x.

**Conditions**
Requires option 5-SREMBD or SUP5-SREMBD.

**Group**
Bus

**Syntax**
BUS:B<x>:I2C:DATa:SOUrce {CH<x> | CH<x>_D<x> | MATH<x> | REF<x> | REF<x>_D<x>} 
BUS:B<x>:I2C:DATa:SOUrce?

**Related Commands**
BUS:B<x>:I2C:CLOCK:SOUrce
BUS:B<x>:I2C:RWINADDR
**Arguments**

- **CH<x>** specifies an analog channel to use as the I2C SDA source.
- **CH<x>_D<x>** specifies a digital channel to use as the I2C SDA source.
- **MATH<x>** specifies a math waveform to use as the I2C SDA source.
- **REF<x>** specifies a reference waveform to use as the I2C SDA source.
- **REF<x>_D<x>** specifies a digital reference waveform as the data source waveform for the specified I2C bus.

**Examples**

- `BUS:B1:I2C:DATA:SOURCE CH1_D5` sets the I2C SDA source to CH1_D5.

**BUS:B<x>:I2C:DATa:THReshold**

This command sets or queries the I2C Data (SDA) source threshold for the specified bus. The bus is specified by `x`.

**Conditions**

Requires option 5-SREMBD or SUP5-SREMBD.

**Group**

Bus

**Syntax**

- `BUS:B<x>:I2C:DATa:THReshold <NR3>`
- `BUS:B<x>:I2C:DATa:THReshold?`

**Arguments**

- **B<x>** is the number of the bus.
- **<NR3>** is the I2C Data (SDA) source threshold for the specified bus.

**Examples**

- `BUS:B1:I2C:DATa:THReshold 50.0e-3` sets the threshold to 50 mV.
- `BUS:B1:I2C:DATa:THReshold?` might return `:BUS:B1:I2C:DATA:THRESHOLD 0.0E+0` indicating the threshold is set to 0 V.

**BUS:B<x>:I2C:RWINADDR**

This command sets or queries the manner in which seven-bit I2C addresses are represented in the busform display of the specified bus. The bus is specified by `x`.

**Conditions**

Requires option 5-SREMBD or SUP5-SREMBD.
Commands listed in alphabetical order

Group Bus

Syntax

BUS:B<x>:I2C:RWINADDR {0|1}
BUS:B<x>:I2C:RWINADDR?

Related Commands

BUS:B<x>:I2C:CLOCK:SOURce
BUS:B<x>:I2C:DATA:SOURce

Arguments

B<x> is the number of the bus.

0 displays seven-bit slave addresses as integers in the range of 0 to 127, with the state of the R/W* bit from the LSB of the slave address byte. For example, the slave address byte of 0b10100101 is displayed as the value 0x52 R.

1 displays the entire slave address byte as a number, with the R/W* signal as its LSB (bit 0) and the slave address in bits 7..1. For example, the slave address byte of 0b10100101 is displayed as the value 0xA5 R.

Examples

BUS:B1:I2C:RWINADDR 0 displays seven-bit slave addresses as integers in the range of 0 to 127.

BUS:B1:I2C:RWINADDR? might return BUS:B1:I2C:RWINADDR 1, indicating that the entire slave address byte is displayed as a number, with the R/W* signal as its LSB (bit 0) and the slave address in bits 7..1.

BUS:B<x>:LABel:COLor

This command sets or queries the color of the specified bus label. The bus is specified by x.

Group Bus

Syntax

BUS:B<x>:LABel:COLor <QString>
BUS:B<x>:LABel:COLor?

Arguments

B<x> is the number of the bus.

<QString> is the bus label color. To return the color to the default color, send an empty string as in this example: :BUS:B1:LABEL:COLOR "".
Examples  
BUS:B1:LAbel:COLor "#FF0000" sets the label color to red. 
BUS:B1:LAbel:COLor? might return :BUS:B1:LAbel:COLOR "#FF0000" indicating the color is red.

**BUS:B<x>:LAbel:FONT:BOLD**

This command sets or queries the bold state of the specified bus label. The bus is specified by x.

**Group**  Bus

**Syntax**  
BUS:B<x>:LAbel:FONT:BOLD {ON|OFF|1|0}
BUS:B<x>:LAbel:FONT:BOLD?

**Arguments**  
B<x> is the number of the bus.
ON displays the label in bold font.
OFF does not display the label in bold font.
1 displays the label in bold font.
0 does not display the label in bold font.

**Examples**  

**BUS:B<x>:LAbel:FONT:ITALic**

This command sets or queries the italic state of the specified bus label. The bus is specified by x.

**Group**  Bus

**Syntax**  
BUS:B<x>:LAbel:FONT:ITALic {ON|OFF|1|0}
BUS:B<x>:LAbel:FONT:ITALic?

**Arguments**  
B<x> is the number of the bus.
ON displays the label in italic font.
OFF does not display the label in italic font.
1 displays the label in italic font.
0 does not display the label in italic font.

**Examples**

```plaintext
1 indicating the font is italic.
```

**BUS:B<x>:LABEL:FONT:SIZE**

This command sets or queries the font size of the specified bus label. The bus is specified by x.

**Group**
Bus

**Syntax**

```plaintext
BUS:B<x>:LABEL:FONT:SIZE <NR1>
BUS:B<x>:LABEL:FONT:SIZE?
```

**Arguments**

- `<x>` is the number of the bus.
- `<NR1>` is the font size.

**Examples**

```plaintext
BUS:B1:LABEL:FONT:SIZE 10 sets the font size to 10 points.
```

**BUS:B<x>:LABEL:FONT:TYPE**

This command sets or queries the font type of the specified bus label, such as Arial or Times New Roman. The bus is specified by x.

**Group**
Bus

**Syntax**

```plaintext
BUS:B<x>:LABEL:FONT:TYPE <QString>
BUS:B<x>:LABEL:FONT:TYPE?
```
Arguments  

B<x> is the number of the bus. <QString> is the specified font type. Available fonts include: DejaVu Sans, DejaVu Sans Mono, DejaVu Serif, Frutiger LT Std, Monospace, Sans Serif, Serif, Ubuntu, Ubuntu Condensed, and Ubuntu Mono.

Examples  

BUS:B1:LABel:FONT:TYPE Monospace selects a monospace font.


**BUS:B<x>:LABel:FONT:UNDERline**

This command sets or queries the underline state of the specified bus label. The bus is specified by x.

Group  

Bus

Syntax  

BUS:B<x>:LABel:FONT:UNDERline {ON|OFF|1|0}  
BUS:B<x>:LABel:FONT:UNDERline?

Arguments  

B<x> is the number of the bus.  

ON displays the label in underlined font.  

OFF does not display the label in underlined font.  

1 displays the label in underlined font.  

0 does not display the label in underlined font.

Examples  

BUS:B:B:LABEL:FONT:UNDERline ON turns on underline font.  

**BUS:B<x>:LABel:name**

This command sets or queries the label for the specified bus. The bus is specified by x.

Group  

Bus
**Syntax**

BUS:B<x>:LABEL:name <QString>

BUS:B<x>:LABEL:name?

**Related Commands**

BUS:B<x>:TYPE

**Arguments**

B<x> is the number of the bus.

<QString> is an alphanumeric string of text enclosed in quotes. The text string is limited to 30 characters. It contains the text label information for the bus.

**Examples**

BUS:B1:LABEL:NAME "TEST" sets the waveform label for the bus B1 to Test.

BUS:B1:LABEL:NAME? might return :BUS:B1:LABEL:NAME "BUS 1", indicating that the waveform label for the bus B1 is set to "Bus 1".

**BUS:B<x>:LABEL:XPOS**

This command sets or queries the x-position of the specified bus label. The bus is specified by x.

**Group**

Bus

**Syntax**

BUS:B<x>:LABEL:XPOS <NR3>

BUS:B<x>:LABEL:XPOS?

**Arguments**

B<x> is the number of the bus.

<NR3> is the x-position, in pixels relative to the left edge of the screen of the specified bus label.

**Examples**

BUS:B1:LABEL:XPOS 90 set the x position to 90.

BUS:B1:LABEL:XPOS? might return :BUS:B1:LABEL:XPOS 45.0000 indicating the x position is 45.0 pixels to the right of the left edge of the display.

**BUS:B<x>:LABEL:YPOS**

This command sets or queries the y-position of the specified bus label. The bus is specified by x.

**Group**

Bus
Commands listed in alphabetical order

**Syntax**

BUS:B<x>:LABel:YPOS <NR3>
BUS:B<x>:LABel:YPOS?

**Arguments**

B<x> is the number of the bus.

<NR3> is the y-position, in pixels relative to the baseline of the waveform, of the specified bus label.

**Examples**

BUS:B1:LABel:YPOS 1.0E1 set the y position to 10.0.
BUS:B1:LABel:YPOS? might return :BUS:B1:LABEL:YPOS 0.0E+0 indicating the y position is 0.0 pixels from the baseline of the waveform.

**BUS:B<x>:LIN:BITRate**

This command sets or queries the LIN bus bit rate. The bus number is specified by x. If you select Custom, use BUS:B<x>:LIN:BITRate:CUSTom to set the bit rate.

**Conditions**

Requires option SR-AUTO.

**Group**

Bus

**Syntax**

BUS:B<x>:LIN:BITRate
{RATE10K|RATE1K|RATE19K|RATE2K|RATE4K|RATE9K|CUSTom}
BUS:B<x>:LIN:BITRate?

**Related Commands**

BUS:B<x>:LIN:BITRate:CUSTom

**Arguments**

B<x> is the number of the bus.

Arguments are the available bit rates.

**Examples**

BUS:B1:LIN:BITRate Rate4k sets the bit rate to 4 kb.
BUS:B1:LIN:BITRate? might return :BUS:B1:BITRATE RATE19K, indicating that the bit rate is set to 19 kb.

**BUS:B<x>:LIN:BITRate:CUSTom**

This command sets or queries LIN custom bit rate for the specified bus. The bus is specified by x.
Commands listed in alphabetical order

**Conditions**
Requires option SR-AUTO.

**Group**
Bus

**Syntax**

```
BUS:B<x>:LIN:BITRate:CUSTOM <NR1>
BUS:B<x>:LIN:BITRate:CUSTom?
```

**Related Commands**

`BUS:B<x>:LIN:BITRate`

**Arguments**

- `B<x>` is the number of the bus.
- `<NR1>` is the LIN custom bit rate for the specified bus.

**Examples**

```
BUS:B1:LIN:BITRate:CUSTom 9000 sets the bit rate to 9,000.
BUS:B1:LIN:BITRate:CUSTom? might return
:BUS:B1:LIN:BITRate:CUSTom 10000 indicating the bit rate is set to 10,000.
```

**BUS:B<x>:LIN:IDFORMAT**

This command sets or queries LIN bus identifier format for the specified bus. The bus number is specified by `x`.

**Conditions**
Requires option SR-AUTO.

**Group**
Bus

**Syntax**

```
BUS:B<x>:LIN:IDFORMAT {NOPARity|PARity}
BUS:B<x>:LIN:IDFORMAT?
```

**Arguments**

- `B<x>` is the number of the bus.
- `NOParity` specifies an id format that includes parity.
- `PARity` specifies an id format that separates parity.

**Examples**

```
BUS:B1:LIN:IDFORMAT PARITY set the id format to parity.
```
**BUS:B<x>:LIN:POLarity**

This command sets or queries the LIN source polarity for the specified bus. The bus number is specified by x.

**Conditions**
Requires option SR-AUTO.

**Group**
Bus

**Syntax**

```
BUS:B<x>:LIN:POLarity {INVerted|NORmal}
BUS:B<x>:LIN:POLarity?
```

**Arguments**

- **B<x>** is the number of the bus.
- **INVerted** specifies inverted polarity.
- **NORmal** specifies normal polarity.

**Examples**

```
BUS:B1:LIN:POLarity Inverted sets the polarity to inverted.
```

**BUS:B<x>:LIN:SAMPLEpoint**

Specifies the LIN sample point, for the specified LIN bus. The bus is specified by x.

**Conditions**
Requires option SR-AUTO.

**Group**
Bus

**Syntax**

```
BUS:B<x>:LIN:SAMPLEpoint <NR1>
BUS:B<x>:LIN:SAMPLEpoint?
```

**Arguments**

- **B<x>** is the number of the bus.
- **<NR1>** is a percentage that represents the point at which to sample during each bit period.
Examples

BUS:B1:LIN:SAMPLEPOINT 10 sets the sample point to 10% of the bit period
indicating that the sample point is set to 50% of the bit period

BUS:B<x>:LIN:SOUrce

This command sets or queries the LIN data source for the specified bus. The bus number is specified by x.

Conditions
Requires option SR-AUTO.

Group
Bus

Syntax
BUS:B<x>:LIN:SOURCE
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:LIN:SOUrce?

Arguments
B<x> is the number of the bus.
Arguments specify the available sources.

Examples
BUS:B1:LIN:SOURCE MATH4 sets the source to MATH 4
the source is set to channel 1.

BUS:B<x>:LIN:SOUrce:THReshold

This command sets or queries the LIN source threshold for the specified bus. The bus is specified by x.

Conditions
Requires option SR-AUTO.

Group
Bus

Syntax
BUS:B<x>:LIN:SOUrce:THReshold <NR3>
BUS:B<x>:LIN:SOUrce:THReshold?
Arguments

B<x> is the number of the bus.

<NR3> the LIN source threshold for the specified bus.

Examples

BUS:B1:LIN:SOURCE:THRESHold 50.0e-3 sets the threshold to 50 mV.


:BUS:B1:LIN:SOURCE:THRESHold 0.0E+0 indicating the threshold is 0.0 V.

BUS:B<x>:LIN:STANDARD

This command sets or queries the LIN bus standard for the specified bus. The bus number is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

BUS:B<x>:LIN:STANDARD {MIXed|V1X|V2X}

BUS:B<x>:LIN:STANDARD?

Arguments

B<x> is the number of the bus.

MIXed specifies both versions 1.x and 2.x of the LIN standard.

V1X specifies version 1.x of the LIN standard.

V2X specifies version 2.x of the LIN standard.

Examples

BUS:B1:LIN:STANDARD V1X sets the standard to version 1.x.


BUS:B<x>:MIL1553B:LOWTHRESHold

This command sets or queries the MIL-STD-1553 lower threshold for the specified bus. The bus is specified by <x>.

Conditions

Requires option SR-AERO.
Commands listed in alphabetical order

**GROUP**

**Bus**

**Syntax**

BUS:B<x>:MIL1553B:LOWTHRESHOLD <NR3>

**Related Commands**

BUS:B<x>:MIL1553B:SOURce

**Arguments**

B<x> is the number of the bus.

<NR3> is the MIL-STD-1553 lower threshold for the specified bus.

**Examples**

BUS:B3:MIL1553B:LOWTHRESHOLD -200e-3 sets the Bus 3 MIL-STD-1553 lower threshold to -200 mV.


**BUS:B<x>:MIL1553B:POLarity**

This command sets or queries the source polarity for the specified MIL-STD-1553 bus. The bus is specified by x.

**Conditions**

Requires option SR-AERO.

**Group**

Bus

**Syntax**

BUS:B<x>:MIL1553B:POLarity {NORMal|INVERTed}

BUS:B<x>:MIL1553B:POLarity?

**Arguments**

B<x> is the number of the bus.

NORMal specifies normal polarity.

INVERTed specifies inverted polarity.

**Examples**


**BUS:B<x>:MIL1553B:RESPonsetime:MAXimum**

This command sets or queries the maximum response time to a valid command issued for the specified MIL-STD-1553 bus. The bus is specified by x.

**Conditions**
Requires option SR-AERO.

**Group**
Bus

**Syntax**

```plaintext
BUS:B<x>:MIL1553B:RESPonsetime:MAXimum <NR3>

BUS:B<x>:MIL1553B:RESPonsetime:MAXimum?
```

**Related Commands**

`BUS:B<x>:MIL1553B:RESPonsetime:MINimum`

**Arguments**

- **B<x>** is the number of the bus.
- `<NR3>` is a floating point number that specifies the maximum response time, in seconds.

**Examples**

```plaintext
BUS:B1:MIL1553B:RESPonsetime:MAXimum 15.0E-6 specifies the maximum response time to a valid command received to be 15.0 microseconds.

```

---

**BUS:B<x>:MIL1553B:RESPonsetime:MINimum**

This command sets or queries the minimum response time to a valid command issued for the specified MIL-STD-1553 bus. The bus is specified by x.

**Conditions**
Requires option SR-AERO.

**Group**
Bus

**Syntax**

```plaintext
BUS:B<x>:MIL1553B:RESPonsetime:MINimum <NR3>

BUS:B<x>:MIL1553B:RESPonsetime:MINimum?
```

**Related Commands**

`BUS:B<x>:MIL1553B:RESPonsetime:MAXimum`
Arguments

<NR3> is a floating point number that specifies the minimum response time, in seconds.

Examples

BUS:B1:MIL1553B:RESPONSETIME:MINIMUM 5.0E-6 specifies the minimum response time to a valid command received to be 5.0 microseconds.


BUS:B<x>:MIL1553B:SOUrce

This command sets or queries the source for the specified MIL-STD-1553 bus. The bus is specified by x.

Conditions

Requires option SR-AERO.

Group

Bus

Syntax

BUS:B<x>:MIL1553B:SOUrce {CH<x>|Math<x>|REF<x>}
BUS:B<x>:MIL1553B:SOUrce?

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source waveform for the MIL-STD-1553 bus.

Math<x> specifies a math waveform as the source waveform for the MIL-STD-1553 bus.

REF<x> specifies a reference waveform as the source waveform for the MIL-STD-1553 bus.

Examples

BUS:B1:MIL1553B:SOURcE CH1 sets channel 1 as the source for the MIL-STD-1553 bus.

BUS:B1:MIL1553B:SOURcE? might return :BUS:B1:MIL1553B:SOURcE MATH2 indicating that the source is set to MATH2.

BUS:B<x>:MIL1553B:THRESHold

This command sets or queries the MIL-STD-1553 upper threshold for the specified bus. The bus is specified by x.
Commands listed in alphabetical order

**Conditions**
Requires option SR-AERO.

**Group**
Bus

**Syntax**

```
BUS:B<x>:MIL1553B:THRESHold <NR3>
BUS:B<x>:MIL1553B:THRESHold
```

**Related Commands**

```
BUS:B<x>:MIL1553B:SOUrce
```

**Arguments**

- **B<x>** is the number of the bus.
- **<NR3>** is the MIL-STD-1553 upper threshold for the specified bus.

**Examples**

```
BUS:B3:MIL1553B:THRESHOLD 2.5 sets the Bus 3 MIL-STD-1553 upper threshold to 2.5 V.
```

**BUS:B<x>:PARallel:ALLTHResholds**

This command sets or queries a threshold value for sources for the parallel bus. Use the **BUS:B<x>:PARallel:ALLTHResholds:APPlY** command to set the thresholds to this value. The bus is specified by **x**.

**Group**
Bus

**Syntax**

```
BUS:B<x>:PARallel:ALLTHResholds <NR3>
```

**Related Commands**

```
BUS:B<x>:PARallel ALLTHResholds APPlY
```

**Arguments**

- **B<x>** is the number of the bus.
- **<NR3>** is the source threshold.

**Examples**

```
BUS:B4:PARALLEL:ALLTHRESHOLDS 1.0 sets the threshold of all the sources in parallel Bus 4 to 1.0 V.
```
BUS:B3:PARALLEL:ALLTHRESHOLDS? might return
BUS:B3:PARALLEL:ALLTHRESHOLDS 500.0000E-3 indicates the threshold for
all sources in parallel Bus 3 is currently set to 500.0 mV.

**BUS:B<x>:PARallel:ALLTHresholds:APPly (No Query Form)**

This command sets all of the data source thresholds to the value set by
BUS:B<x>:PARallel:ALLTHresholds for the parallel bus. The bus is specified
by x.

**Group**  
Bus

**Syntax**  
BUS:B<x>:PARallel:ALLTHresholds:APPly

**Related Commands**  
BUS:B<x>:PARallel:ALLTHresholds

**Examples**  
BUS:B1:PARallel:ALLTHresholds:APPly sets all data source thresholds to
the value set by BUS:B<x>:PARallel:ALLTHresholds.

**BUS:B<x>:PARallel:BIT<n>SOUrce**

This command sets or queries the specified bit source for specified parallel bus.
The bus is specified by x. The bit is specified by n and is an integer in the range of
1 to 64.

**Group**  
Bus

**Syntax**  
BUS:B<x>:PARallel:BIT<n>SOUrce  
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>|NONE}

**Related Commands**  
BUS:B<x>:PARallel:BIT<n>SOUrce:THReshold

**Arguments**  
B<x> is the number of the bus.
CH<x> is the specified bit source.
CH<x>_D<x> is the specified bit source.
MATH<x> is the specified bit source.
REF<x> is the specified bit source.
REF<x>_D<x> specifies a digital reference waveform as the bit<x> source waveform for the specified parallel bus. NONE disables the bit source.

Examples

BUS:B1:PARALLEL:BIT1SOURCE CH1 sets the bit 1 source to channel 1.
BUS:B1:PARALLEL:BIT1SOURCE? might return :BUS:B1:PARALLEL:BIT1SOURCE CH1_D0 indicating the bit1 source is CH1_D0.

BUS:B<x>:PARALLEL:BIT<n>SOURCE:THRESHold

This command sets or queries the specified bit source threshold for the specified parallel bus. The bus is specified by x. The bit is specified by n and is an integer in the range of 1 to 64.

Group Bus

Syntax BUS:B<x>:PARALLEL:BIT<n>SOURCE:THRESHold <NR3>

Related Commands BUS:B<x>:PARALLEL BIT<n>SOURCE

Arguments B<x> is the number of the bus. <NR3> is the specified bit source threshold for the specified parallel bus.

Examples BUS:B3:PARALLEL:BIT2SOURCE:THRESHOLD 1.0 sets the threshold of bit source 2 of parallel Bus 3 to 1.0 V.

BUS:B<x>:PARALLEL:CLOCK:EDGE

This command sets or queries the clock edge for the parallel bus. The bus is specified by x.

Group Bus
Syntax

```
BUS:B<x>:PARallel:CLOCk:EDGE {FALLING|RISING|EITHER}
BUS:B<x>:PARallel:CLOCk:EDGE?
```

Related Commands

```
BUS:B<x>:PARallel:CLOCKSOURce
```

Arguments

- **B<x>** is the number of the bus.
- **FALLING** decodes on the falling edge of the clocked parallel bus signal.
- **RISING** decodes on the rising edge of the clocked parallel bus signal.
- **EITHER** decodes on the rising or falling edge of the clocked parallel bus signal.

Examples

```
BUS:B1:PARALLEL:CLOCK:EDGE FALLING sets the decoding to happen on the falling edge of its clocked parallel bus B1.

```

**BUS:B<x>:PARallel:CLOCK:ISCLOCKED**

This command determines whether the bus operates in a clocked or asynchronous fashion. The bus is specified by **x**.

Group

Bus

Syntax

```
BUS:B<x>:PARallel:CLOCK:ISCLOCKED {OFF|ON|NR1>}
BUS:B<x>:PARallel:CLOCK:ISCLOCKED?
```

Arguments

- **B<x>** is the number of the bus.
- **OFF** argument specifies an asynchronous bus.
- **ON** argument specifies a clocked bus.
- **<NR1> = 0** specifies an asynchronous bus, any other value specifies a clocked bus.

Examples

```
BUS:B1:PARALLEL:CLOCK:ISCLOCKED 0 sets the bus to operate asynchronously.

```
BUS:B<x>:PARallel:CLOCKsOURce

This command sets or queries the Parallel clock bit source for the specified bus. The bus is specified by x.

Group  Bus

Syntax  BUS:B<x>:PARallel:CLOCKsOURce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>|NONE}
BUS:B<x>:PARallel:CLOCKsOURce?

Related Commands  BUS:B<x>:PARallel:CLOCK:ISClocked

Arguments  B<x> is the number of the bus.
CH<x> specifies an analog FlexChannel to use as the bus clock source.
CH<x>_D<x> specifies a digital channel on a specified FlexChannel to use as the bus clock source.
MATH<x> specifies the math channel to use as the bus clock source.
REF<x> specifies the reference channel to use as the bus clock source.
REF<x>_D<x> specifies a digital reference waveform as the clock source waveform for the specified parallel bus.
NONE specifies the reference channel to use as the bus clock source.

Examples  BUS:B1:PARALLEL:CLOCK:SOURCE CH3_D6 sets the Parallel clock source for the bus B1 to D6 of FlexChannel 3.

BUS:B<x>:PARallel:CLOCKsOURce:THReshold

This command sets or queries the clock source threshold for the parallel bus. The bus is specified by x.

Group  Bus
Syntax

BUS:B<x>:PARallel:CLOCKSOUrce:THReshold <NR3>
BUS:B<x>:PARallel:CLOCKSOUrce:THReshold?

Related Commands

BUS:B<x>:PARallel:CLOCK:ISCLOCKED

Arguments

<NR3> is the clock bit source threshold for the parallel bus.

Examples

BUS:B4:PARALLEL:CLOCKSOUrce:THRESHOLD 1.5 sets the threshold of the clock source of parallel Bus 4 to 1.5 V.

BUS:B3:PARALLEL:CLOCKSOUrce:THRESHOLD? might return :BUS:B3:PARALLEL:CLOCKSOUrce:THRESHOLD 750.00000E-3 indicates the threshold of the clock source of parallel Bus 3 is currently set to 750.0 mV.

**BUS:B<x>:RS232C:BITRate**

This command sets or queries the RS-232C bit rate for bus<x>, where the bus number is specified by <x>. If you select Custom, use BUS:B<x>:RS232C:BITRate:Custom to set the bit rate.

Conditions

Requires option SR-COMP.

Group

Bus

Syntax

BUS:B<x>:RS232C:BITRate
{CUSTOM|RATE300|RATE1K|RATE2K|RATE9K|RATE19K|RATE38K|RATE115K|RATE921K}
BUS:B<x>:RS232C:BITRate?

Arguments

B<x> is the number of the bus.

Arguments are the available bit rates.

Examples


**BUS:B<x>:RS232C:BITRate:CUSTom**

This command sets or queries the RS-232C custom bit rate for the specified bus. The bus is specified by x.

**Conditions**
Requires option SR-COMP.

**Group**
Bus

**Syntax**

```
BUS:B<x>:RS232C:BITRate:CUSTom <NR1>
BUS:B<x>:RS232C:BITRate:CUSTom?
```

**Related Commands**

- **B<x>** is the number of the bus.
- **BUS:B<x>:RS232C:BITRate**

**Arguments**

- **<NR1>** is the custom bit rate.

**Examples**

```

```
Examples

BUS:B1:RS232C:DATABITS 8 sets the data bits to 8 for the RS-232C bus B1.


BUS:B<x>:RS232C:DELIMITER

This command sets or queries the RS-232C string delimiter on bus <x>, where the bus number is specified by <x>. This command only applies when Packet view is turned On.

Conditions
Requires option SR-COMP.

Group
Bus

Syntax
BUS:B<x>:RS232C:DELIMITER {NULl|CR|LF|SPace|XFF}  
BUS:B<x>:RS232C:DELIMITER?

Related Commands
BUS:B<x>:RS232C:DISPLAYMODE

Arguments

B<x> is the number of the bus.

NULl specifies NULL (0x00) delimiting value for a packet.

CR specifies CR (0xD) delimiting value for a packet.

LF specifies LF (0xA) delimiting value for a packet.

XFF specifies XFF (0xFF) delimiting value for a packet.

SPACE specifies SPACE delimiting value for a packet.

Examples

BUS:B1:RS232C:DELIMITER LF sets the delimiter to 0x0A for the RS-232C bus B1.


BUS:B<x>:RS232C:DISPLAYMODE

This command sets or queries the RS-232C display mode for the specified bus. The bus is specified by x.
Commands listed in alphabetical order

**Conditions**
Requires option SR-COMP.

**Group**
Bus

**Syntax**

```
BUS:B<x>:RS232C:DISPLAYMODE {FRAME|PACKET}
BUS:B<x>:RS232C:DISPLAYMODE?
```

**Related Commands**
`BUS:B<x>:RS232C:DELIMITER`

**Arguments**
`B<x>` is the number of the bus.

- **FRAME** displays each frame as a single entity.
- **PACKET** displays a group of frames terminated with a single frame defined by the `BUS:B<x>:RS232C:DELIMITER` command.

**Examples**

```
BUS:B1:RS232C:DISPLAYMODE FRAME sets the display mode for the RS-232C bus B1 to Frame.
```

**BUS:B<x>:RS232C:PARITY**

This command sets or queries the RS-232C parity for bus `<x>`, where the bus number is specified by `<x>`.

**Conditions**
Requires option SR-COMP.

**Group**
Bus

**Syntax**

```
BUS:B<x>:RS232C:PARITY {NONE|EVEN|ODD}
BUS:B<x>:RS232C:PARITY?
```

**Arguments**
`B<x>` is the number of the bus.

- **NONE** specifies no parity.
- **EVEN** specifies even parity.
- **ODD** specifies odd parity.
Examples

BUS:B1:RS232C:PARITY ODD sets the parity for the RS-232C bus B1 to odd.


BUS:B<x>:RS232C:POLarity

This command sets or queries the RS-232C source polarity for bus <x>, where the bus number is specified by <x>.

Conditions
Requires option SR-COMP

Group
Bus

Syntax
BUS:B<x>:RS232C:POLarity {NORMAL|INVERTed}
BUS:B<x>:RS232C:POLarity?

Arguments

B<x> is the number of the bus.

NORMAL sets the RS-232C bus polarity to positive.

INVERTed sets the RS-232C bus polarity to negative.

Examples


BUS:B<x>:RS232C:SOURce

This command sets or queries the RS-232C source for bus <x>, where the bus number is specified by <x>.

Conditions
Requires option SR-COMP

Group
Bus

Syntax
BUS:B<x>:RS232C:SOURce {CH<x>|CH<x>_D<x>|REF<x>|MATH<x>|REF<x>_D<x>}
BUS:B<x>:RS232C:SOURce?
Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel to use as the RS-232C source.

CH<x>_D<x> specifies a digital channel of a specified FlexChannel to use for the RS-232C source.

MATH<x> specifies a math channel to use for the RS-232C source.

REF<x> specifies a reference channel to use for the RS-232C source.

REF<x>_D<x> specifies a digital reference waveform as the source waveform for the specified RS-232C bus.

Examples

BUS:B1:RS232C:SOURCE CH1_D0 sets the source for the RS-232C bus B1 to D0 of FlexChannel 1.


BUS:B<x>:RS232C:SOUrce:THReshold

This command sets or queries the RS-232C source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-COMP.

Group

Bus

Syntax

BUS:B<x>:RS232C:SOUrce:THReshold <NR3>

BUS:B<x>:RS232C:SOUrce:THReshold?

Arguments

B<x> is the number of the bus.

<NR3> is the RS-232C source threshold for the specified bus.

Examples

BUS:B1:RS232C:SOUrce:THReshold 50.0e-3 sets the threshold to 50 mV.

BUS:B1:RS232C:SOUrce:THReshold? might return :BUS:B1:RS232C:SOUrce:THReshold 0.0E+0 indicating the threshold is set to 0.0 V.
**BUS:B<x>:SENT:CHANWidth**

This command sets or queries SENT fast channel bit widths for the specified bus.

**Conditions**
Requires option SRAUTOSEN.

**Group**
Bus

**Syntax**
BUS:B<x>:SENT:CHANWidth  
{TWELVE|FOURTEEN|SIXTEEN}  
BUS:B<x>:SENT:CHANWidth?

**Related Commands**
BUS:B<x>:SENT:NUMCHANnel

**Arguments**

B<x> is the number of the bus waveform.

TWELVE sets both Fast Channel 1 and Fast Channel 2 to 12 bits wide.

FOURTEEN sets Fast Channel 1 to 14 bits and Fast Channel 2 to 10 bits.

SIXTEEN sets Fast Channel 1 to 16 bits and Fast Channel 2 to eight bits.

**Examples**

BUS:B1:SENT:CHANWIDTH TWELVE sets the channel width of both Fast Channel 1 and Fast Channel 2 to twelve bits.

BUS:B1:SENT:CHANWIDTH? might return :BUS:B1:SENT:CHANWIDTH SIXTEEN, which indicates that the Fast Channel 1 width is 16 and the Fast Channel 2 width is eight.

---

**BUS:B<x>:SENT:NIBBLECount**

This command sets or queries SENT data nibbles for the specified bus.

**Conditions**
Requires option SRAUTOSEN.

**Group**
Bus

**Syntax**
BUS:B<x>:SENT:NIBBLECount {THREE|FOUR|SIX}  
BUS:B<x>:SENT:NIBBLECount?
Arguments  

B<x> is the number of the bus waveform. 
THREE specifies regular data with 3 nibbles 
FOUR specifies regular data with 4 nibbles 
SIX specifies regular data with 6 nibbles 

Examples  

BUS:B1:SENT:NIBBLECOUNT THREE sets the data nibble count to three. 

**BUS:B<x>:SENT:NUMCHANnel**

This command sets or queries SENT fast data channels for the specified bus.

Conditions  

Requires option SRAUTOSEN.

Group  

Bus

Syntax  

BUS:B<x>:SENT:NUMCHANnel {ONE|TWO} 
BUS:B<x>:SENT:NUMCHANnel?

Arguments  

B<x> is the number of the bus waveform. 
ONE specifies a SENT bus with one fast channel. 
TWO specifies a SENT bus with two fast channels.

Examples  

BUS:B1:SENT:NUMCHANnel TWO sets the number of SENT channels to two. 
BUS:B1:SENT:NUMCHANnel? might return :BUS:B1:SENT:NUMCHANnel ONE to indicate that the number of channels is set to one.

**BUS:B<x>:SENT:PAUSEPULSe**

This command sets or queries SENT pause pulse for the specified bus.

Conditions  

Requires option SRAUTOSEN.

Group  

Bus
Syntax

BUS:B<x>:SENT:PAUSEPULSE {NO|YES}
BUS:B<x>:SENT:PAUSEPULSE?

Arguments

B<x> is the number of the bus waveform.
NO specifies no pause pulse.
YES specifies a pause pulse is used.

Examples

BUS:B1:SENT:PAUSEPULSE NO sets the pause pulse value to indicate no pause pulse.

BUS:B<x>:SENT:POLARITY

This command sets or queries SENT Idle State signal polarity for the specified bus.

Conditions

Requires option SRAUTOSEN.

Group

Bus

Syntax

BUS:B<x>:SENT:POLARITY {INVerted|NORmal}
BUS:B<x>:SENT:POLARITY?

Arguments

B<x> is the number of the bus waveform.
INVerted specifies inverted polarity.
NORmal specifies normal polarity.

Examples

BUS:B1:SENT:POLARITY INVERTED sets the idle state of the bus to inverted polarity.

BUS:B<x>:SENT:SLOW

This command sets or queries the SENT slow channel configuration for the specified bus.
<table>
<thead>
<tr>
<th>Conditions</th>
<th>Requires option SRAUTOSEN.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Bus</td>
</tr>
<tr>
<td>Syntax</td>
<td>`BUS:B&lt;x&gt;:SENT:SLOW {NONE</td>
</tr>
<tr>
<td>Arguments</td>
<td><code>B&lt;x&gt;</code> is the number of the bus waveform.&lt;br&gt;<code>NONE</code> specifies no slow channel configured.&lt;br&gt;<code>ENHANCED4</code> specifies Enhanced 4 slow channel configuration.&lt;br&gt;<code>ENHANCED8</code> specifies Enhanced 8 slow channel configuration.&lt;br&gt;<code>SHORT</code> specifies short slow channel configuration.</td>
</tr>
<tr>
<td>Examples</td>
<td><code>BUS:B1:SENT:SLOW SHORT</code> sets the SENT bus to use the short slow channel configuration.&lt;br&gt;<code>BUS:B1:SENT:SLOW?</code> might return <code>:BUS:B1:SENT:SLOW NONE</code> to indicate that the SENT bus is set to have no slow channel.</td>
</tr>
</tbody>
</table>

**BUS:B<x>:SENT:SOUrce**

This command sets or queries the SENT DATA source for the specified bus.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Requires option SRAUTOSEN.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Bus</td>
</tr>
<tr>
<td>Syntax</td>
<td>`BUS:B&lt;x&gt;:SENT:SOUrce {CH&lt;x&gt;</td>
</tr>
<tr>
<td>Arguments</td>
<td><code>B&lt;x&gt;</code> is the number of the bus waveform.&lt;br&gt;<code>CH&lt;x&gt;</code> specifies an analog channel as the clock source waveform for the audio bus.&lt;br&gt;<code>CH&lt;x&gt;_D&lt;x&gt;</code> specifies a digital channel as the clock source waveform for the specified audio bus.&lt;br&gt;<code>Math&lt;x&gt;</code> specifies a math waveform as the clock source waveform for the audio bus.</td>
</tr>
</tbody>
</table>

---

2-194 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
REF<x> specifies a reference waveform as the clock source waveform for the audio bus.

REF<x>_D<x> specifies a digital reference waveform as the clock source waveform for the specified audio bus.

**Examples**

```plaintext
BUS:B1:SENT:SOURCE CH1_D1 sets D1 of channel 1 as the source for the SENT bus.

BUS:B1:SENT:SOURCE? might return :BUS:B1:SENT:SOURCE CH1 indicating that the source is set to channel 1
```

**BUS:B<x>:SENT:THRESHold**

This command sets or queries the SENT DATA source threshold for the specified bus.

**Conditions**

Requires option SRAUTOSEN.

**Group**

Bus

**Syntax**

```plaintext
BUS:B<x>:SENT:THRESHold <NR3>
BUS:B<x>:SENT:THRESHold?
```

**Arguments**

*B<x>* is the number of the bus waveform.

<NR3> sets the data source threshold value in volts.

**Examples**

```plaintext
BUS:B1:SENT:THRESHold 500.0E-3 sets the Bus 1 source threshold to 500.0 mV.

BUS:B1:SENT:THRESHold? might return :BUS:B1:SENT:THRESHold 2.5000 to indicate the Bus 1 source threshold is 2.5 V.
```

**BUS:B<x>:SENT:TICKTIME**

This command sets or queries the SENT bus Clock Tick parameter for the specified bus.

**Conditions**

Requires option SRAUTOSEN.
### BUS:B<x>:SENT:TICKTIME

This command sets or queries the SENT bus Tick Tolerance percent parameter for the specified bus.

#### Conditions
Requires option SRAUTOSEN.

#### Syntax
- `BUS:B<x>:SENT:TICKTIME <NR3>`
- `BUS:B<x>:SENT:TICKTIME?`

#### Arguments
- `B<x>` is the number of the bus waveform.
- `<NR3>` is the SENT clock tick time, in seconds.

#### Examples
- `BUS:B1:SENT:TICKTIME 3.0E-6` sets the SENT bus to clock tick to 3 microseconds.

### BUS:B<x>:SENT:TICKTOLERANCE

This command sets or queries the SENT bus Tick Tolerance percent parameter for the specified bus.

#### Syntax
- `BUS:B<x>:SENT:TICKTOLERANCE <NR3>`
- `BUS:B<x>:SENT:TICKTOLERANCE?`

#### Arguments
- `B<x>` is the number of the bus waveform.
- `<NR3>` is the tick tolerance percentage.

#### Examples
- `BUS:B1:SENT:TICKTOLERANCE 20.0` sets the tick tolerance to 20%.
- `BUS:B1:SENT:TICKTOLERANCE?` might return `:BUS:B1:SENT:TICKTOLERANCE 22.5` to indicate the tolerance is set to 22.5%.

### BUS:B<x>:SPI:BITOrder

This command sets or queries the SPI bit order for the specified bus. The bus is specified by `x`.
Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax \texttt{BUS:B<x>:SPI:BITOrder} \{\texttt{LSB|MSB}\}
\texttt{BUS:B<x>:SPI:BITOrder?}

Related Commands \texttt{BUS:B<x>:SPI:DATa:SIZe}

Arguments \(B<x>\) is the number of the bus waveform.

\texttt{LSB} specifies that each bit becomes the recovered value's new LSB, after shifting previously recovered bits one place to the left. The decoding happens right to left.

\texttt{MSB} specifies that each successive bit from the bus's data line becomes the new MSB of the recovered value, shifting any previously recovered bits one place to the right. The decoding happens left to right.

Examples \texttt{BUS:B1:SPI:BITORDER LSB} sets each bit order for the SPI bus B1 to LSB.
\texttt{BUS:B1:SPI:BITORDER?} might return \texttt{BUS:B1:SPI:BITORDER MSB}, indicating that the bit order for the SPI bus B1 is set to MSB.

\textbf{BUS:B<x>:SPI:CLOCk:POLarity}

This command sets or queries the SPI clock (SCLK) source polarity for the specified bus. The bus is specified by \(x\).

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax \texttt{BUS:B<x>:SPI:CLOCk:POLarity} \{\texttt{FALL|RISE}\}
\texttt{BUS:B<x>:SPI:CLOCk:POLarity?}

Related Commands \texttt{BUS:B<x>:SPI:CLOCk:SOUrce}

Arguments \(B<x>\) is the number of the bus waveform.

\texttt{FALL} sets the clock to the falling edge of the signal.
RISE sets the clock to the rising edge of the signal.

**Examples**

```plaintext
BUS:B1:SPI:CLOCK:POLARITY FALL sets the SPI clock polarity to the falling edge for the bus B1.

```

**BUS:B<x>:SPI:CLOCK:SOURCE**

This command sets or queries the SPI clock (SCLK) source for the specified bus. The bus is specified by x.

**Conditions**
Requires option 5-SREMBD or SUP5-SREMBD.

**Group**
Bus

**Syntax**

```plaintext
BUS:B<x>:SPI:CLOCK:SOURCE
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:SPI:CLOCK:SOURCE?
```

**Related Commands**

**BUS:B<x>:SPI:CLOCK:POLARITY**

**Arguments**

B<x> is the number of the bus waveform.

CH<x> designates an analog channel as the bus SPI clock source.

CH<x>_D<x> designates a digital channel as the bus SPI clock source.

MATH<x> designates a math waveform as the clock source.

REF<x> designates a reference waveform as the clock source.

REF<x>_D<x> specifies a digital reference waveform as the clock source waveform for the specified SPI bus.

**Examples**

```plaintext
BUS:B1:SPI:CLOCK:SOURCE CH5 sets the SPI clock source for the bus B1 to CH5.

```
BUS:B<x>:SPI:CLOCK:THReshold

This command sets or queries the SPI Clock (SCLK) source threshold for the specified bus. The bus is specified by x.

**Conditions**
Requires option 5-SREMBD or SUP5-SREMBD.

**Group**
Bus

**Syntax**
BUS:B<x>:SPI:CLOCK:THReshold <NR3>
BUS:B<x>:SPI:CLOCK:THReshold?

**Arguments**
B<x> is the number of the bus waveform.
<NR3> is the SPI Clock (SCLK) source threshold for the specified bus.

**Examples**
BUS:B1:SPI:CLOCK:THReshold 50.0e-3 sets the threshold to 50.0 mV.
BUS:B1:SPI:CLOCK:THReshold? might return :BUS:B1:SPI:CLOCK:THRESHOLD 0.0E+0, indicating the threshold is set to 0.0 V.

BUS:B<x>:SPI:DATa:POLarity

This command sets or queries the SPI Data (Data) source polarity for the bus number specified by x.

**Conditions**
Requires option 5-SREMBD or SUP5-SREMBD.

**Group**
Bus

**Syntax**
BUS:B<x>:SPI:DATa:POLarity {HIGH|LOW}
BUS:B<x>:SPI:DATa:POLarity?

**Arguments**
B<x> is the number of the bus waveform.
HIGH sets the SPI data polarity to active high.
LOW sets the SPI data polarity to active low.
Examples

BUS:B4:SPI:DATA:POLARITY HIGH sets the SPI data polarity for the bus B4 to active high.


**BUS:B<x>:SPI:DATA:SIZE**

This command sets or queries the number of bits per word for the specified SPI bus. The bus is specified by x.

**Conditions**

Requires option 5-SREMBD or SUP5-SREMBD.

**Group**

Bus

**Syntax**

BUS:B<x>:SPI:DATA:SIZE <NR1>
BUS:B<x>:SPI:DATA:SIZE?

**Arguments**

B<x> is the number of the bus waveform.

<NR1> is the data size for the specified bus. The minimum value is 2 and maximum is 32.

**Examples**

BUS:B1:SPI:DATA:SIZE 7 sets the data size for SPI bus B1 to seven bits per word.


**BUS:B<x>:SPI:DATA:SOURCE**

This command sets or queries the SPI Data (Data) source for the bus number specified by x.

**Conditions**

Requires option 5-SREMBD or SUP5-SREMBD.

**Group**

Bus

**Syntax**

BUS:B<x>:SPI:DATA:SOURCE
{CH<x> | CH<x>_D<x> | MATH<x> | REF<x> | REF<x>_D<x>}
BUS:B<x>:SPI:DATA:SOURCE?
Arguments

B<x> is the number of the bus waveform.

CH<x> designates an analog channel as the data source for the specified SPI bus.

CH<x>_D<x> designates an digital channel as the bus SPI clock source.

MATH<x> designates a math waveform as the data source.

REF<x> designates a reference waveform as the data source.

REF<x>_D<x> specifies a digital reference waveform as the data source waveform for the specified SPI bus.

Examples


BUS:B<x>:SPI:DATa:THReshold

This command sets or queries the SPI Data (Data) source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD.

Group

Bus

Syntax

BUS:B<x>:SPI:DATa:THReshold <NR3>
BUS:B<x>:SPI:DATa:THReshold?

Arguments

B<x> is the number of the bus waveform.

<NR3> is the SPI Data (SDA) source threshold for the specified bus.

Examples

BUS:B1:SPI:DATa:THReshold 50.0e-3 sets the threshold to 50.0 mV.

BUS:B1:SPI:DATa:THReshold? might return :BUS:B1:SPI:DATa:THReshold 0.0E+0 indicating the threshold is set to 0.0 V.

BUS:B<x>:SPI:FRAMING

This command sets or queries the SPI framing setting for the specified bus. The bus number is specified by x.
Commands listed in alphabetical order

**Conditions**
Requires option 5-SREMBD or SUP5-SREMBD.

**Group**
Bus

**Syntax**

```
BUS:B<x>:SPI:FRAMING {IDLE|SS}
BUS:B<x>:SPI:FRAMING?
```

**Arguments**

- **B<x>** is the number of the bus waveform.
- **IDLE** specifies IDLE SPI framing.
- **SS** specifies SS SPI framing.

**Examples**

```
BUS:B1:SPI:FRAMING IDLE sets the SPI framing to IDLE.
BUS:B1:SPI:FRAMING? might return :BUS:B1:SPI:FRAMING SS, indicating that the SPI framing is set to SS.
```

**BUS:B<x>:SPI:IDLETime**

This command sets or queries the SPI idle time for the specified bus. The bus is specified by x.

**Conditions**
Requires option 5-SREMBD or SUP5-SREMBD.

**Group**
Bus

**Syntax**

```
BUS:B<x>:SPI:IDLETime <NR3>
BUS:B<x>:SPI:IDLETime?
```

**Arguments**

- **B<x>** is the number of the bus waveform.
- **<NR3>** specifies the SPI idle time.

**Examples**

```
BUS:B1:SPI:IDLETime 0.000004 sets the idle time to 4 μs.
BUS:B1:SPI:IDLETime? might return :BUS:B1:SPI:IDLETIME 5.0000E-6, indicating that the idle time is set to 5 μs.
```
BUS:B<x>:SPI:SELECT:POLarity

This command sets or queries the SPI Slave Select (SS) polarity for the specified bus. The bus is specified by x.

Conditions
Requires option 5-SREMBD or SUP5-SREMBD.

Group
Bus

Syntax
BUS:B<x>:SPI:SELECT:POLarity {LOW|HIGH}
BUS:B<x>:SPI:SELECT:POLarity?

Related Commands
BUS:B<x>:SPI:SELECT:SOURce

Arguments
B<x> is the number of the bus waveform.
LOW sets an active low polarity.
HIGH sets an active high polarity.

Examples
BUS:B1:SPI:SELECT:POLARITY HIGH sets the SPI Slave Select polarity for the bus B1 to active high.

BUS:B<x>:SPI:SELECT:SOURce

This command sets or queries the SPI Slave Select (SS) source for the specified bus. The bus is specified by x.

Conditions
Requires option 5-SREMBD or SUP5-SREMBD.

Group
Bus

Syntax
BUS:B<x>:SPI:SELECT:SOURce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:SPI:SELECT:SOURce?

Related Commands
BUS:B<x>:SPI:SELECT:POLarity
Arguments

B<\text{x}> is the number of the bus waveform.

CH<\text{x}>, CH<\text{x}>_D<\text{x}>, MATH<\text{x}>, REF<\text{x}>, REF<\text{x}>_D<\text{x}>

Examples

BUS:B1:SPI:SELECT:SOURCE MATH3 sets the SPI Slave Select source for
the bus B1 to MATH3.

CH3, indicating that the SPI Slave Select source for the bus B1 is set to CH3.

**BUS:B<\text{x}>:SPI:SELECT:THRESHold**

This command sets or queries the SPI Select (SS) source threshold for the
specified bus. The bus is specified by x.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD.

Group

Bus

Syntax

BUS:B<\text{x}>:SPI:SELECT:THRESHold <NR3>
BUS:B<\text{x}>:SPI:SELECT:THRESHold?

Arguments

B<\text{x}> is the number of the bus waveform.

<NR3> is the SPI Select (SS) source threshold for the specified bus.

Examples

BUS:B1:SPI:SELECT:THRESHold 50.0e-3 sets the threshold to 50.0 mV.

BUS:B1:SPI:SELECT:THRESHold? might return :BUS:B1:SPI:SELECT:THRESHOLD 0.0E+0 indicating the threshold is set to
0.0 V.
BUS:B<x>:SPMI:SCLk:SOUrce

This command sets or queries the SPMI Clock (SCLK) source for the specified bus.

Conditions
Requires option SRPM

Group
Bus

Syntax
BUS:B<x>:SPMI:SCLk:SOUrce {CH<x>|CH<x>_D<x>|Math<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:SPMI:SCLk:SOUrce

Related Commands
BUS:B<x>:SPMI:SDATa:SOUrce

Arguments
B<x> is the number of the bus waveform.
CH<x> specifies an analog channel as the clock source waveform for the SPMI bus.
CH<x>_D<x> specifies a digital channel and bit as the clock source waveform for the specified SPMI bus.
Math<x> specifies a math waveform as the clock source waveform for the specified SPMI bus.
REF<x> specifies a reference waveform as the clock source waveform for the specified SPMI bus.
REF<x>_D<x> specifies a digital reference waveform and bit as the clock source waveform for the specified SPMI bus.

Examples
BUS:B1:SPMI:SCLk:SOUrce CH1_D1 sets D1 of channel 1 as the clock source waveform for the SPMI bus B1.

BUS:B<x>:SPMI:SCLk:THReshold

This command sets or queries the SPMI Clock (SCLK) source threshold for the specified bus.

Conditions
Requires option SRPM
Commands listed in alphabetical order

**Group**  Bus

**Syntax**  
BUS:B<x>:SPMI:SCLk:THReshold  <NR3>
BUS:B<x>:SPMI:SCLk:THReshold?

**Related Commands**  TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition

**Arguments**  
B<x> is the number of the bus waveform.
<NR3> is the clock (SCLK) source threshold value for the specified SPMI bus.

**Examples**  
BUS:B3:SPMI:SCLK:THRESHOLD 500.0E-3 sets the Bus3 clock source threshold to 500.0 mV.
BUS:B1:SPMI:SCLK:THRESHOLD? might return :BUS:B1:SPMI:SCLK:THRESHOLD 250.0E-3 to indicate Bus1 clock threshold is set to 250.0 mV.

**BUS:B<x>:SPMI:SDATa:SOUrce**

This command sets or queries the SPMI Data (SDATA) source for the specified bus.

**Conditions**  Requires option SRPM

**Group**  Bus

**Syntax**  
BUS:B<x>:SPMI:SDATa:SOUrce {CH<x>|CH<x>_Dx>|Math<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:SPMI:SDATa:SOUrce

**Related Commands**  BUS:B<x>:SPMI:SCLk:SOUrce

**Arguments**  
B<x> is the number of the bus waveform.
CH<x> specifies an analog channel as the data source waveform for the SPMI bus.
CH<x>_Dx> specifies a digital channel and bit as the data source waveform for the specified SPMI bus.
Math<x> specifies a math waveform as the data source waveform for the specified SPMI bus.
REF<x> specifies a reference waveform as the data source waveform for the specified SPMI bus.

REF<x>_D<x> specifies a digital reference waveform and bit as the data source waveform for the specified SPMI bus.

**Examples**

BUS:B4:SPMI:SDATA:SOURCE CH1_D2 sets bit D2 of digital channel 1 as the data source for the SPMI bus B4.


### BUS:B<x>:SPMI:SDATA:THREshold

This command sets or queries the SPMI Data (SDATA) source threshold for the specified bus.

**Conditions**

Requires option SRPM

**Group**

Bus

**Syntax**

BUS:B<x>:SPMI:SDATA:THREshold <NR3>

BUS:B<x>:SPMI:SDATA:THREshold?

**Related Commands**

TRIGger:<A|B>:BUS:B<x>:SPMI:CONDition

**Arguments**

B<x> is the number of the bus waveform.

<NR3> is the data (SDATA) source threshold value for the specified SPMI bus.

**Examples**

BUS:B2:SPMI:SDATA:THRESHOLD 400.0E-3 sets Bus2 data source threshold to 400.0 mV.

BUS:B1:SPMI:SDATA:THRESHOLD? might return :BUS:B1:SPMI:SDATA:THRESHOLD 220.0E-3 to indicate that Bus1 data source threshold is set to 220.0 mV.

### BUS:B<x>:TPYe

This command sets or queries the bus type or standard for the specified bus. The bus is specified by x.
Group  
Bus

Syntax  
BUS:B<x>:TYPE {ARINC429|Audio|CAN|ETHERnet|FLEXRAY|I2C|LIN|MIL1553B|PARallel|RS232C|SENT|SPI|SPMI|USB} 
BUS:B<x>:TYPE?

Arguments  
B<x> is the number of the bus waveform.
ARINC429 specifies the ARINC 429 avionics serial bus.
Audio specifies an audio bus.
CAN specifies a Controller Area Network bus.
ETHERnet specifies the Ethernet bus.
FLEXRAY specifies a FlexRay bus.
I2C specifies the Inter-IC bus.
LIN specifies a Local Interconnect Network bus.
MIL1553B specifies the MIL-STD-1553 avionics serial bus.
PARallel specifies a parallel bus.
RS232C specifies the RS-232 Serial bus.
SENT specifies the Single Edge Nibble Transmission (SENT) automotive serial bus.
SPI specifies the Serial Peripheral Interface bus.
SPMI specifies a System Power Management Interface bus.
USB specifies the Universal Serial Bus.

Examples  
BUS:B1:TYPE I2C sets the bus B1 type to I2C.
BUS:B1:TYPE? might return :BUS:B1:TYPE SPI, indicating that the bus B1 type is set to SPI and that the bus behavior is governed by the :BUS:B1:SPI:xxxx commands.

BUS:B<x>:USB:BITRate

This command sets or queries the USB data rate for bus <x>, where the bus number is specified by <x>.

Conditions  
Requires option SR-USB2.
Commands listed in alphabetical order

**GROUP: Bus**

**Syntax**

`BUS:B<x>:USB:BITRate {FULL|HIGH|LOW}`

`BUS:B<x>:USB:BITRate?`

**Arguments**

*B<x>* is the number of the bus waveform.

FULL indicates the bit rate is 12 Mbps.

HIGH indicates the bit rate is 480 Mbps.

LOW indicates the bit rate is 1.5 Mbps.

**Examples**

`BUS:B1:USB:BITRATE FULL` sets the bit rate to 12 Mbps.

`BUS:B1:USB:BITRATE?` might return `:BUS:B1:USB:BITRATE LOW`, indicating that the bit rate is 1.5 Mbps.

**BUS:B<x>:USB:DATAMINUSTHRESHold**

This command sets or queries the USB D- source threshold for the specified bus. The bus is specified by *x*.

**Conditions**

Requires option SR-USB 2.

**GROUP: Bus**

**Syntax**

`BUS:B<x>:USB:DATAMINUSTHRESHold <NR3>`

`BUS:B<x>:USB:DATAMINUSTHRESHold?`

**Arguments**

*B<x>* is the number of the bus waveform.

*<NR3>* is the Minus threshold.

**Examples**

`BUS:B1:USB:DATAMINUSTHRESHold 50.0e-3` sets the threshold to 50.0 mV.

`BUS:B1:USB:DATAMINUSTHRESHold?` might return `:BUS:B1:USB:DATAMINUSTHRESHOLD 0.0E+0` indicating the threshold is set to 0.0 V.
**BUS:B<x>:USB:DATAPLUSTHRESHold**

This command sets or queries the USB D+ source threshold for the specified bus. The bus is specified by x.

**Conditions**
Requires option SR-USB 2.

**Group**
Bus

**Syntax**

```
BUS:B<x>:USB:DATAPLUSTHRESHold <NR3>
BUS:B<x>:USB:DATAPLUSTHRESHold?
```

**Arguments**

- `B<x>` is the number of the bus waveform.
- `<NR3>` is the Plus threshold.

**Examples**

```
BUS:B1:USB:DATAPLUSTHRESHold 50.0e-3 sets the threshold to 50.0 mV.
BUS:B1:USB:DATAPLUSTHRESHold? might return
:BUS:B1:USB:DATAPLUSTHRESHOLD 0.0E+0 indicating the threshold is set to 0.0 V.
```

**BUS:B<x>:USB:LOWTHRESHold**

This command sets or queries the USB data source threshold for the specified bus when the signal type is differential. The bus is specified by x.

**Conditions**
Requires option SR-USB 2.

**Group**
Bus

**Syntax**

```
BUS:B<x>:USB:LOWTHRESHold <NR3>
BUS:B<x>:USB:LOWTHRESHold?
```

**Arguments**

- `B<x>` is the number of the bus waveform.
- `<NR3>` is the Low threshold.
Examples

BUS:B1:USB:LOWTHRESHOLD 50.0e-3 sets the threshold to 50.0 mV.

BUS:B1:USB:LOWTHRESHOLD? might return :BUS:B1:USB:LOWTHRESHOLD 0.0E+0 indicating the threshold is set to 0.0 V.

**BUS:B<x>:USB:SIGNALTYpe**

This command sets or queries the USB signal type for the specified bus. The bus is specified by x.

**Conditions**
Requires option SR-USB 2.

**Group**
Bus

**Syntax**
BUS:B<x>:USB:SIGNALTYpe {SINGLE|DIFF}
BUS:B<x>:USB:SIGNALTYpe?

**Arguments**
B<x> is the number of the bus waveform.
SINGLE specifies single-ended signals.
DIFF specifies differential signals.

**Examples**
BUS:B1:USB:SIGNALTYpe SINGLE specifies single-ended signals.

**BUS:B<x>:USB:SOUrce**

This command sets or queries the USB data source when the signal type is differential for bus <x>. The bus number is specified by <x>.

**Conditions**
Requires option SR-USB 2.

**Group**
Bus

**Syntax**
BUS:B<x>:USB:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:USB:SOUrce?
Related Commands

BUS:B<x>:USB:SOUrce:DMINus
BUS:B<x>:USB:SOUrce:DPLUs

Arguments

B<x> is the number of the bus waveform.
CH<x> specifies an analog channel as the data source for the specified USB bus.
MATH<x> specifies a math channel as the data source for the specified USB bus.
REF<x> specifies a reference waveform as the data source.

Examples

BUS:B1:USB:SOURCE D2 sets the USB data source for bus B1 to D2.

BUS:B<x>:USB:SOUrce:DMINus

This command sets or queries the USB D- (SDATAMINUS) source for bus <x> when the signal type is single ended. The bus number is specified by <x>.

Conditions

Requires option SR-USB 2.

Group

Bus

Syntax

BUS:B<x>:USB:SOUrce:DMINus
{CH<x> | CH<x>_D<x> | MATH<x> | REF<x> | REF<x>_D<x>} 
BUS:B<x>:USB:SOUrce:DMINus?

Related Commands

BUS:B<x>:USB:SOUrce
BUS:B<x>:USB:SOUrce:DPLUs
BUS:B<x>:USB:SIGNALTYpe

Arguments

B<x> is the number of the bus waveform.
CH<x> specifies an analog channel as the D- source for the specified USB bus.
CH<x>_D<x> specifies a digital channel as the D- source for the specified USB bus.
MATH<x> specifies a math channel as the D- source for the specified USB bus.
REF<x> specifies a reference waveform as the source.
REF<x>_D<x> specifies a digital reference waveform as the clock source waveform for the specified USB bus.

**Examples**

BUS:B1:USB:SOURCE:DMINUS CH2 sets the USB Data Source for D- input to CH2.


**BUS:B<x>:USB:SOURce:DPLUS**

This command sets or queries the USB dataPlus (SDATAPLUS) source for the specified bus when the signal type is single ended. The bus is specified by x.

**Conditions**

Requires option SR-USB 2.

**Group**

Bus

**Syntax**

BUS:B<x>:USB:SOURce:DPLUS

{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}

BUS:B<x>:USB:SOURce:DPLUS?

**Related Commands**

BUS:B<x>:USB:SOURce

BUS:B<x>:USB:SOURce:DMINus

BUS:B<x>:USB:SIGNALTYpe

**Arguments**

B<x> is the number of the bus waveform.

CH<x> specifies an analog channel as the D+ source for the specified USB bus.

CH<x>_D<x> specifies a digital channel as the D+ source for the specified USB bus.

MATH<x> specifies a math channel as the D+ source for the specified USB bus.

REF<x> specifies a reference waveform as the source.

REF<x>_D<x> specifies a digital reference waveform as the clock source waveform for the specified USB bus.

**Examples**

BUS:B1:USB:SOURce:DPLUS CH2 sets the USB Data Source for D+ input to CH2.

**BUS:B<x>:USB:THRESHold**

This command sets or queries the USB DATA source High threshold for the specified bus when the signal source is differential. The bus is specified by x.

- **Conditions** Requires option SR-USB2.
- **Group** Bus
- **Syntax** 
  ```
  BUS:B<x>:USB:THRESHold <NR3>  
  BUS:B<x>:USB:THRESHold?
  ```
- **Arguments** 
  - B<x> is the number of the bus waveform.
  - <NR3> is the USB DATA source High threshold for the specified bus.
- **Examples** 
  - BUS:B1:USB:THRESHold 50.0E-3 sets the threshold to 50.0 mV.
  - BUS:B1:USB:THRESHold? might return :BUS:B1:USB:THRESHOLD 0.0E+0 indicating the threshold is set to 0.0 V.

**BUS:DELETE (No Query Form)**

This command deletes the specified bus.

- **Group** Bus
- **Syntax** 
  ```
  BUS:DELETE <QString>
  ```
- **Arguments** 
  - <QString> specifies the bus to delete and is of the form "B<NR1>", where <NR1> is ≥1.
- **Examples** 
**BUS:LIST? (Query Only)**

This query returns a comma separated list of all currently defined buses.

- **Group**: Bus
- **Syntax**: `BUS:LIST?`
- **Returns**: Returns all currently defined buses.
- **Examples**: `BUS:LIST?` might return `:BUS:LIST B1,B4` indicating the bus 1 and bus 4 are defined.

**BUSTABLE:ADDNew (No Query Form)**

Adds the specified bus table. Argument is of the form "TABLE<NR1>", where `<NR1>` is ≥1.

- **Group**: Bus
- **Syntax**: `BUSTABLE:ADDNew <QString>`
- **Arguments**: `<QString>` is a quoted string that is the name of the new bus table.
- **Examples**: `BUSTABLE:ADDNEW "Table1"` adds bus table Table1.

**BUSTABLE:DELETE (No Query Form)**

Deletes the specified bus table. Argument is of the form "TABLE<NR1>", where `<NR1>` is ≥1.

- **Group**: Bus
- **Syntax**: `BUSTABLE:DELETE <QString>`
- **Arguments**: `<QString>` is a quoted string that is the name of the bus table to delete.
Examples  BUSTABLE:DELETE "Table1" deletes bus table Table1.

BUSTABLE:LIST? (Query Only)

This query lists all currently defined bus tables.

Group  Bus

Syntax  BUSTABLE:LIST?

Returns  Returns a list of all currently defined bus tables.

Examples  BUSTABLE:LIST? might return :BUSTABLE:LIST TABLE1 indicating TABLE1 is currently the only defined bus table.

BUSY? (Query Only)

This query-only command returns the status of the instrument. This command allows you to synchronize the operation of the instrument with your application program.

Group  Status and Error

Syntax  BUSY?

Related Commands  *OPC
                 *WAI

Returns  <NR1> = 0 means that the instrument is not busy processing a command whose execution time is extensive.
         <NR1> = 1 means that the instrument is busy processing Commands that Generate an OPC Message (See Table 2-46.).

Examples  BUSY? might return :BUSY 1, indicating that the instrument is currently busy.
**CAL? (Query Only)**

This query-only command starts signal path calibration (SPC) and returns the status upon completion.

**NOTE.** When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 5 minutes per channel which means a total of 40 minutes on an 8-channel model. Any remote command that performs an action on the oscilloscope is also disabled until the SPC is complete.

**Group** Calibration

**Syntax** *CAL?*

**Returns**
- 0 indicates SPC passed.
- -1 indicates SPC failed or did not complete.

**Examples**

*CAL?* starts the signal path calibration and returns the status upon completion.

**CALibrate? (Query Only)**

This query returns the status of signal path calibration.

**NOTE.** When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes, which can take several minutes.

**Group** Calibration

**Syntax** CALibrate?

**Examples**

**CALibrate:INTERNal (No Query Form)**

This command (no query form) starts the signal path calibration (SPC) of the instrument. You can use the `CALibrate:INTERNal:STATus?` query to return the current status of the signal path calibration of the instrument.

**NOTE.** When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 5 minutes per channel which means a total of 40 minutes on an 8-channel model. Any remote command that performs an action on the oscilloscope is also disabled until the SPC is complete.

---

**Group**  
Calibration

**Syntax**  
`CALibrate:INTERNal`

**Related Commands**  
`CALibrate:INTERNal:STATus?`

**Examples**  
`CALIBRATE:INTERNAL` starts the signal path calibration of the instrument.

---

**CALibrate:INTERNal:STARt (No Query Form)**

This command (no query form) starts the signal path calibration (SPC) of the analog channels. This command is the same as the `CALibrate:INTERNal` command. You can use the `CALibrate:INTERNal:STATus?` query to return the current status of the signal path calibration of the instrument.

**NOTE.** When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 5 minutes per channel which means a total of 40 minutes on an 8-channel model. Any remote command that performs an action on the oscilloscope is also disabled until the SPC is complete.

---

**Group**  
Calibration

**Syntax**  
`CALibrate:INTERNal:STARt`

**Related Commands**  
`CALibrate:INTERNal:STATus?`
Examples CALIBRATE:INTERNAL:START starts the signal path calibration.

CALibrate:INTERNAL:STATus? (Query Only)
This query-only command returns the current status of the signal path calibration.

NOTE. When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 5 minutes per channel which means a total of 40 minutes on an 8-channel model. Any remote command that performs an action on the oscilloscope is also disabled until the SPC is complete.

Group Calibration

Syntax CALibrate:INTERNAL:STATus?

Related Commands *CAL?

Returns This query will return one of the following:
- INIT indicates the instrument has not had signal path calibration run. The instrument may need to be readjusted at the Tektronix service center.
- PASS indicates that the signal path calibration completed successfully.
- FAIL indicates that the signal path calibration did not complete successfully.

Examples CALIBRATE:INTERNAL:STATUS? might return PASS, indicating that the current status of the signal path calibration is that the signal path calibration completed successfully.

CALibrate:PWRUpstatus? (Query Only)
This query-only command returns the current status of the power-up calibration.

Group Calibration

Syntax CALibrate:PWRUpstatus?
Returns  This query will return one of the following:
- 0 Indicating the power-up calibration failed.
- 1 Indicating the power-up calibration passed.

Examples  CALIBRATE:PWRUPSTATUS? might return 0, indicating that the power-up calibration failed.

CH<n>:PRObe:STATus? (Query Only)

Reads the probe unsigned integer error value.

Conditions  Requires a probe that supports the relevant error messages.

Group  Vertical

Syntax  CH<n>:PRObe:STATus?

Returns  Returns an integer number that represents the sum total of binary error bits B0 – B15. The error bits are not displayed; they are concatenated into the integer value. The following is a list of the error for each bit:
- B0 – Probe disabled
- B1 – Jaws open
- B2 – Over range
- B3 – Probe temperature out of limits
- B4 – Degauss needed
- B5 – Probe tip missing
- B6 – Probe tip failed
- B7 – Probe tip not supported
- B8 through B15 – Reserved

Examples  CH4:PROBE:STATus? might return CH4:PROBE:STATus 0, indicating that the probe is not reporting any errors.

CH4:PROBE:STATus? might return CH4:PROBE:STATus 20, indicating that the probe is reporting an over range error and a Degauss needed error.
Commands listed in alphabetical order

CH4:PROBE:STATus? might return CH4:PROBE:STATus 2, indicating that the probe is reporting an open jaws error.

CH<x>? (Query Only)

This query-only command returns the vertical parameters for the specified channel. The channel is specified by x.

Group   Vertical

Syntax   CH<x>?

Examples   CH1? might return the following vertical parameters for channel 1: CH1:BANDWIDTH 1.0000E+09; COUPLING DC; DESKEW 0.0000E+00; OFFSET 0.0000E+00; POSITION 0.0000E+00; SCALE 5.0000E-01; PROBCAL INIT; PROBE: GAIN 1.0000E+00; RESISTANCE 1.0000E+06; UNITS "V"; ID: TYPE "1X"; SERNUMBER "N/A"; CH1: PROBE: FUNC: EXTATTEN 1.0000E+00; EXTUNITS "None"; CH1: LABEL: NAME ""; XPOS 5; YPOS 5.

CH<x>:BANwidth

This command sets or queries the selectable low-pass bandwidth limit filter of the specified channel. The channel is specified by x.

The query form of this command always returns the approximate realized bandwidth of the channel.

Available arguments depend upon the instrument and the attached accessories.

Group   Vertical

Syntax   CH<x>:BANwidth {<NR3>|FULL}      CH<x>:BANwidth?

Arguments   CH<x> is the channel number.
<NR3> is the desired bandwidth. The instrument rounds this value to an available bandwidth using geometric rounding and then uses this value to set the upper bandwidth.
**FULL** disables any optional bandwidth limiting. The specified channel operates at its maximum bandwidth.

**Examples**

CH1:BANDWIDTH 20 sets the bandwidth of Channel 1 to 20 MHz.

CH2:BANDWIDTH?, might return :CH2:BANDWIDTH 500.0000E+06, indicating that there is bandwidth limiting on Channel 2.

**CH<x>::CLIPping? (Query Only)**

Queries whether the specified channel’s input signal is clipping (exceeding) the channel A/D converter range. The channel is specified by x.

**Group** Vertical

**Syntax** CH<x>::CLIPping?

**Related Commands**

CH<x>::SCAle

CH<x>::POSition

**Returns**

This query will return one of the following:

0 indicates the channel is not clipping.

1 indicates the channel is clipping.

**Examples**

CH4:CLIPping? might return CH4:CLIPping? 1, indicating that the input signal range is exceeding (clipping) the channel 4 A/D converter range.

**CH<x>::COUPling**

This command sets or queries the input coupling setting for the specified analog channel. The channel is specified by x.

**NOTE.** *The available arguments depend on the attached accessories.*

**Group** Vertical
Syntax  \texttt{CH<x>:COUPLing \{AC|DC|DCREJ\}}  \\
\hspace{1cm} \texttt{CH<x>:COUPLing?} \\

Arguments  \texttt{CH<x>} is the channel number. \\
\hspace{1cm} AC sets the specified channel to AC coupling. \\
\hspace{1cm} DC sets the specified channel to DC coupling. \\
\hspace{1cm} DCREject sets DC Reject coupling when probes are attached that support that feature. \\

Examples  \texttt{CH2:COUPLING AC} sets Channel 2 coupling to AC. \\
\hspace{1cm} \texttt{CH3:COUPLING?} might return \texttt{:CH3:COUPLING DC}, indicating that Channel 3 is set to DC coupling. \\

\textbf{CH<x>:DESKew} \\
This command sets or queries the horizontal deskew time for the specified channel. The channel is specified by \texttt{x}. \\

Group  Vertical \\

Syntax  \texttt{CH<x>:DESKew <NR3>}  \\
\hspace{1cm} \texttt{CH<x>:DESKew?} \\

Arguments  \texttt{CH<x>} is the channel number. \\
\hspace{1cm} <NR3> is the deskew time for this channel, ranging from -125 ns to +125 ns with a resolution of 40 ps. Out-of-range values are clipped. \\

Examples  \texttt{CH4:DESKew 5.0E-9} sets the deskew time for Channel 4 to 5 ns. \\
\hspace{1cm} \texttt{CH2:DESKew?} might return \texttt{:CH2:DESKew 2.0000E-09}, indicating that the deskew time for Channel 2 is set to 2 ns. \\

\textbf{CH<x>:LABel:COLor} \\
This command sets or queries the color of the specified channel label. The channel is specified by \texttt{x}. \\

Group  Vertical
Commands listed in alphabetical order

Syntax

**CH<x>:LABEL:COLOR** <QString>

**Arguments**

*CH<x>* is the channel number.

*<QString>* is the label color. To return the color to the default color, send an empty string as in this example: :CH5:LABEL:COLOR "".

**Examples**

CH2:LABEL:COLOR "#FFFF00" sets the label color to yellow.

CH2:LABEL:COLOR? might return :CH2:LABEL:COLOR "#FF0000" indicating the color is red.

**CH<x>:LABEL:FONT:BOLD**

This command sets or queries the bold state of the specified channel label. The channel is specified by x.

**Group**

Vertical

**Syntax**

**CH<x>:LABEL:FONT:BOLD** \{ON|OFF|<NR1}\}

**Arguments**

*CH<x>* is the channel number.

OFF argument turns off bold font.

ON argument turns on bold font.

<NR1> = 0 turns off bold font; any other value turns on bold font.

**Examples**

CH2:LABEL:FONT:BOLD OFF turns off the bold font.


**CH<x>:LABEL:FONT:ITALiC**

This command sets or queries the italic state of the specified channel label. The channel is specified by x.

**Group**

Vertical

**Syntax**

**CH<x>:LABEL:FONT:ITALiC** \{ON|OFF|<NR1}\}
Arguments

CH<x> is the channel number.
OFF argument turns off italic font.
ON argument turns on italic font.
<NR1> = 0 turns off italic font; any other value turns on italic font.

Examples

CH2:LABel:FONT:ITALic ON set the font to italic.

CH<x>:LABel:FONT:SIZE

This command sets or queries the font size of the specified channel label. The channel is specified by x.

Group

Vertical

Syntax

CH<x>:LABel:FONT:SIZE <NR1>

Arguments

CH<x> is the channel number.
<NR1> is the font size.

Examples

CH2:LABel:FONT:SIZE 16 sets the font size to 16 point size.

CH<x>:LABel:FONT:TYPE

This command sets or queries the font type of the specified channel label, such as Arial or Times New Roman. The channel is specified by x.

Group

Vertical

Syntax

CH<x>:LABel:FONT:TYPE <QString>

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
Arguments

- CH<x> is the channel number.
- <QString> is the specified font type.

Examples

CH2:LABEL:FONT:TYPE "Monospace" sets the font to a mono space font.


CH<x>:LABEL:FONT:UNDERline

This command sets or queries the underline state of the specified channel label. The channel is specified by x.

Group
Vertical

Syntax

CH<x>:LABEL:FONT:UNDERline {ON|OFF|<NR1>}

Arguments

- CH<x> is the channel number.
- OFF argument turns off underlined font.
- ON argument turns on underlined font.
- <NR1> = 0 turns off underlined font; any other value turns on underlined font.

Examples

CH2:LABEL:FONT:UNDERline ON sets the font to underlined.


CH<x>:LABel:NAMe

This command sets or queries the label attached to the displayed waveform for the specified channel. The channel is specified by x.

Group
Vertical

Syntax

CH<x>:LABel:NAMe <QString>
CH<x>:LABel:NAMe?
Arguments CH<x> is the channel number.

<QString> is an alphanumeric character string, ranging from 1 through 32 characters in length.

Examples CH2:LABEL:NAME "Pressure" changes the waveform label for the Channel 2 waveform to “Pressure”.

CH3:LABEL:NAME? might return :CH3:LABEL:NAME "Force", indicating that the waveform label for the Channel 3 waveform is “Force”.

CH<x>:LABEL:XPOS

This command sets or queries the X-position of the specified channel label. The channel is specified by x.

Syntax CH<x>:LABEL:XPOS <NR3>
CH<x>:LABEL:XPOS?

Arguments CH<x> is the channel number.

<NR3> is the location (in pixels) where the waveform label for the selected channel is displayed, relative to the left edge of the screen.

Examples CH3:LABEL:XPOS 5 moves the waveform label for Channel 3 so that it begins 5 pixels to the right of the left edge of the screen.

CH2:LABEL:XPOS? might return :CH2:LABEL:XPOS 5, indicating that the waveform label for the Channel 2 currently 5 pixels to the right of the left edge of the screen.

CH<x>:LABEL:YPOS

This command sets or queries the Y-position of the specified channel label. The channel is specified by x.

Group Vertical

Syntax CH<x>:LABEL:YPOS <NR3>
CH<x>:LABEL:YPOS?

Arguments CH<x> is the channel number.

<NR3> is the location (in pixels) where the waveform label for the selected channel is displayed, relative to the left edge of the screen.

Examples CH3:LABEL:YPOS 5 moves the waveform label for Channel 3 so that it begins 5 pixels to the right of the left edge of the screen.

CH2:LABEL:YPOS? might return :CH2:LABEL:YPOS 5, indicating that the waveform label for the Channel 2 currently 5 pixels to the right of the left edge of the screen.
Arguments

CH<x> is the channel number.

<NR3> is the location (in pixels) where the waveform label for the selected channel is displayed, relative to the baseline of the waveform. Positive values are above the baseline and negative values are below.

Examples

CH3:LABEL:YPOS -20 moves the waveform label for the Channel 3 20 pixels below the baseline of the waveform.

CH2:LABEL:YPOS? might return :CH2:LABEL:YPOS 0, indicating that the waveform label for the Channel 2 is currently located at the baseline of the waveform.

CH<x>:OFFSET

This command sets or queries the vertical offset for the specified analog channel.

Group

Vertical

Syntax

CH<x>:OFFSET <NR3>

CH<x>:OFFSET?

Arguments

CH<x> is the channel number.

<NR3> is the offset value for the specified channel.

Examples

CH3:OFFSET 2.0E-3 sets the offset for Channel 3 to 2 mV.

CH4:OFFSET? might return :CH4:OFFSET 1.0000E-03, indicating that the offset for Channel 4 is set to 1 mV.

CH<x>:POSITION

This command sets or queries the vertical position for the specified analog channel.

Group

Vertical

Syntax

CH<x>:POSITION <NR1>
Arguments  CH<x> is the channel number.
           <NR1> is the vertical position for the specified analog channel.

Examples  CH2:POSITION -2.0 sets the position to -2 divisions.
           CH2:POSITION? might return :CH2:POSITION -2.2400 indicating the
position is -2.24 divisions.

CH<x>:PRObe? (Query Only)

This query-only command returns all information concerning the probe that is
attached to the specified channel. The channel is specified by x.

Group  Vertical

Syntax  CH<x>:PRObe?

Related Commands  CH<x>:PROBECal?

Examples  CH2:PROBE? might return :CH2:PROBE:GAIN 1.0000E-01; RESISTANCE
1.0000E+07;UNITS "V";ID:TYPE "10X";SERNUMBER "N/A" for a 10X
probe, indicating that (among other parameters) the attenuation factor for the probe
attached to Channel 2 is 100.0 mV (assuming that probe units are set to volts).

CH<x>:PRObe:AUTOZero (No Query Form)

This command executes the attached probe’s Auto Zero function, for probes that
support this feature. See your probe documentation for more details. The channel
is specified by x.

Group  Vertical

Syntax  CH<x>:PRObe:AUTOZero EXECute

Arguments  CH<x> is the channel number.
           EXECute sets the probe attached to the specified channel to autozero.
**Examples**  
CH1:PROBE:AUTOZERO EXECUTE sets the probe attached to the Channel 1 to autozero.

**CH<x>:PRObe:DEGAUSS (No Query Form)**

This command starts a degauss cycle of the TekVPI probe attached to the specified channel. The channel is specified by x.

**Group**  
Vertical

**Syntax**  
CH<x>:PRObe:DEGAUSS EXECute

**Arguments**  
CH<x> is the channel number.  
EXECute starts the degauss cycle.

**Examples**  
CH1:PROBE:DEGAUSS EXECUTE causes the probe attached to Channel 1 to degauss.

**CH<x>:PRObe:DEGAUSS:STATE? (Query Only)**

This command queries whether the probe attached to the specified channel requires a degauss operation. The channel is specified by x.

**Group**  
Vertical

**Syntax**  
CH<x>:PRObe:DEGAUSS:STATE?

**Returns**  
Required indicates the probe should be degaussed before taking measurements.  
Recommended indicates the measurement accuracy might be improved by degaussing the probe.  
Passed indicates the probe is degaussed.  
Failed indicates the degauss operation failed.  
Inprocess indicates the probe degauss operation is currently in progress.

**Examples**  
CH2:PROBE:DEGAUSS:STATE? might return :CH2:PROBE:DEGAUSS:STATE PASSED, indicating that the probe attached to the Channel 2 is degaussed.
**CH<x>:PRObe:FORCEDRange**

This command sets the attached TekVPI probe to the specified range, or it queries the range of the probe attached to the specified channel. The channel is specified by x.

**Group** Vertical

**Syntax**

```
CH<x>:PRObe:FORCEDRange <NR3>
CH<x>:PRObe:FORCEDRange?
```

**Arguments**

- **CH<x>** is the channel number.
- **<NR3>** specifies the probe dynamic range.

**Examples**

If a TCP0030 current probe is attached to the Channel 1 input,

```
CH1:PROBE:FORCEDRANGE 5.0
```

sets the attached probe to its 5 Ampere range.

```
CH3:PROBE:FORCEDRANGE?
```

might return 

```
:CH1:PROBE:FORCEDRANGE 30.0000
```

indicating that the range of the probe attached to the Channel 3 is set to 30 Amperes.

---

**CH<x>:PRObe:GAIN? (Query Only)**

This query-only command returns the gain factor of the probe that is attached to the specified channel. The channel is specified by x. The gain of a probe is the output divided by the input transfer ratio. For example, a common 10x probe has a gain of 0.1.

**Group** Vertical

**Syntax**

```
CH<x>:PRObe:GAIN?
```

**Examples**

```
CH2:PROBE:GAIN?
```

might return 

```
:CH2:PROBE:GAIN 100.0000E-3
```

indicating that the attached 10X probe delivers 0.1 V to the Channel 2 BNC for every 1.0 V applied to the probe input.

---

**CH<x>:PRObe:ID? (Query Only)**

This query-only command returns the type and serial number of the probe that is attached to the specified channel. The channel is specified by x.
Commands listed in alphabetical order

Group          Vertical

Syntax        CH<x>:PRObe:ID?

Examples      CH2:PROBE:ID? might return CH2:PROBE:ID:TYPE "10X"; SERNUMBER "N/A", indicating that a passive 10X probe of unknown serial number is attached to Channel 2.

CH<x>:PRObe:ID:SERnumber? (Query Only)

This query-only command returns the serial number of the probe that is attached to the specified channel. The channel is specified by x.

**NOTE.** For Level 0 and 1 probes, the serial number will be “N/A”.

Group          Vertical

Syntax        CH<x>:PRObe:ID:SERnumber?

Examples      CH1:PROBE:ID:SERNUMBER? might return CH1:PROBE:ID:SERNUMBER "B010289", indicating that the serial number of the probe attached to Channel 1 is B010289.

CH<x>:PRObe:ID:TYPe? (Query Only)

This query-only command returns the type of probe that is attached to the specified channel. The channel is specified by x.

Group          Vertical

Syntax        CH<x>:PRObe:ID:TYPe?

Examples      CH1:PROBE:ID:TYPe? might return CH1:PROBE:ID:TYPe "TCP0030", indicating that a TCP0030 current probe is attached to Channel 1.
**CH<x>:PRObe:INPUTMode**

Sets or queries the input mode of the probe that is attached to the specified channel.

**Conditions**
Requires a probe with dual inputs.

**Group**
Vertical

**Syntax**
```
CH<x>:PRObe:INPUTMode {A|B|C|D}
CH<x>:PRObe:INPUTMode?
```

**Arguments**
- **CH<x>** is the channel number.
- **A** sets the probe to send single-ended A signals to the instrument.
- **B** sets the probe to send single-ended B signals to the instrument.
- **C** sets the probe to send common-mode signals to the instrument.
- **D** sets the probe to send differential signals to the instrument.

**Examples**
```
CH8:PROBE:INPUTMODE C sets the probe input type to send common mode signals to channel 8.
CH6:PROBE:INPUTMODE? might return :CH6:PROBE:INPUTMODE A, indicating that the input mode of the probe that is attached to the Channel 6 is set to single ended A signals.
```

**CH<x>:PRObe:INPUTMode:AOFFSet**

Sets or queries the A mode offset control of the probe that is attached to the specified channel.

**Conditions**
Requires a probe with dual inputs.

**Group**
Vertical

**Syntax**
```
CH<x>:PRObe:INPUTMode:AOFFSet <NR3>
CH<x>:PRObe:INPUTMode:AOFFSet?
```
Arguments  

CH<x> is the channel number.

<NR3> sets the A mode offset value, in vertical units (V or A).

Examples  

CH2:PROBE:INPUTMODE:AOFFSET 5.0 sets the A mode offset control value of the probe that is attached to the Channel 2 to 5.

CH1:PROBE:INPUTMODE:AOFFSET? might return
CH1:PROBE:INPUTMODE:AOFFSET? 0.0000, indicating that the A mode offset value of the probe that is attached to the channel 1 is set to 0.0.

CH<x>:PROBE:INPUTMode:BOFFSet

Sets or queries the B mode offset control of the probe that is attached to the specified channel.

Conditions  

Requires a probe with dual inputs.

Group  

Vertical

Syntax  

CH<x>:PROBE:INPUTMode:BOFFSet <NR3>

CH<x>:PROBE:INPUTMode:BOFFSet?

Arguments  

CH<x> is the channel number.

<NR3> sets the B mode offset value, in vertical units (V or A).

Examples  

CH2:PROBE:INPUTMODE:BOFFSET 5.0 sets the B mode offset value of the probe that is attached to the Channel 2 to 5.

CH5:PROBE:INPUTMODE:BOFFSET? might return
CH5:PROBE:INPUTMODE:BOFFSET? 1.2500, indicating that the B mode offset value of the probe that is attached to the channel 5 is set to 1.25.

CH<x>:PROBE:INPUTMode:COFFSet

Sets or queries the common mode offset value of the probe that is attached to the specified channel.

Conditions  

Requires a probe with dual inputs.
Commands listed in alphabetical order

**Group**
Vertical

**Syntax**

CH<x>:PRObe:INPUTMode:COFFSet <NR3>
CH<x>:PRObe:INPUTMode:COFFSet?

**Arguments**

CH<x> is the channel number.

<NR3> sets the C (common) mode offset value, in vertical units (V or A).

**Examples**

CH7:PROBE:INPUTMODE:COFFSET 3.2 sets the C mode offset value of the probe that is attached to the Channel 7 to 3.2.

CH5:PROBE:INPUTMODE:COFFSET? might return

CH5:PROBE:INPUTMODE:COFFSET? 1.0250, indicating that the C mode offset value of the probe that is attached to the channel 5 is set to 1.025.

**CH<x>:PRObe:INPUTMode:DOFFSet**

Sets or queries the differential offset value of the probe that is attached to the specified channel.

**Conditions**

Requires a probe with dual inputs.

**Group**
Vertical

**Syntax**

CH<x>:PRObe:INPUTMode:DOFFSet <NR3>
CH<x>:PRObe:INPUTMode:DOFFSet?

**Arguments**

CH<x> is the channel number.

<NR3> sets the D (differential) mode offset value, in vertical units (V or A).

**Examples**

CH7:PROBE:INPUTMODE:DOFFSET 3.2 sets the D mode offset value of the probe that is attached to the Channel 7 to 3.2.

CH5:PROBE:INPUTMODE:DOFFSET? might return

CH5:PROBE:INPUTMODE:DOFFSET? 1.0250, indicating that the D mode offset value of the probe that is attached to the channel 5 is set to 1.025.
Commands listed in alphabetical order

CH<x>:PRObe:RESistance? (Query Only)

This query-only command returns the resistance of the probe that is attached to the specified channel. The channel is specified by x.

Group Vertical

Syntax CH<x>:PRObe:RESistance?

Examples CH2:PROBE:RESISTANCE? might return :CH2:PROBE:RESISTANCE 1.0000E+06, indicating that the input resistance of the probe attached to Channel 2 is 1 MΩ.

CH<x>:PRObe:SET

This command sets or queries aspects of probe accessory user interfaces, for example probe attenuation factors or probe audible over range. The available arguments for this command will vary depending on the accessory you attach to the instrument. The channel is specified by x.

Group Vertical

Syntax CH<x>:PRObe:SET <QString>
CH<x>:PRObe:SET?

Arguments CH<x> is the channel number.

<QString> is a quoted string representing a settable aspect of the attached accessory.

Examples CH6:PRObe:SET "ATTENUATION 10X" sets the probe to 10X attenuation.
CH6:PRObe:SET? might return CH6:PROBE:SET "ATTENUATION 25X" indicating that the probe is set to the selection for 25X attenuation.

CH<x>:PRObe:UNIts? (Query Only)

This query-only command returns a string describing the units of measure for the probe attached to the specified channel. The channel is specified by x.
Commands listed in alphabetical order

**CH<x>:PRObe:UNIts?**

**Syntax**  
CH<x>:PRObe:UNIts?

**Related Commands**  
CH<x>:PROBEFunc:EXTUnits

**Examples**  
CH4:PROBE:UNITS? might return CH4:PROBE:UNITS "V", indicating that the units of measure for the probe attached to Channel 4 are volts.

**CH<x>:PROBECal? (Query Only)**

This query-only command returns the probe calibration state for the specified channel. The channel is specified by x.

**Group**  
Vertical

**Syntax**  
CH<x>:PROBECal?

**Returns**  
This query will return one of the following:
- **Failed** signifies that the probe calibration has failed for the selected channel.
- **Default** signifies that the probe calibration has not yet been run for the selected channel.
- **Passed** signifies the probe calibration has passed for the selected channel.
- **Running** signifies the probe calibration is running.

**Examples**  
CH2:PROBECal? might return CH2:PROBECAL PASSED indicating that the probe calibration has passed for Channel 2.

**CH<x>:PROBECOntrol**

This command sets or queries multirange probe range-control policy preference of the probe that is attached to CH<x>. The channel number is specified by <x>.

**Group**  
Vertical
Commands listed in alphabetical order

Syntax

CH<x>:PROBECControl {AUTO|MANual}
CH<x>:PROBECControl?

Arguments

CH<x> is the channel number.

AUTO sets the values. The probe range is automatically calculated.

MANual allows you to select various valid values for the probe connected to a particular channel.

Examples

CH2:PROBECCONTROL AUTO sets the values and the probe range is automatically calculated.

CH2:PROBECCONTROL? might return :CH2:PROBECCONTROL MANUAL indicating that you can select various valid values for the probe connected to channel 2.

CH<x>:PROBEFunc:EXTAtten

This command is used to specify the attenuation value as a multiplier to the given scale factor on the specified channel. The channel is specified by x.

The query form of this command returns the user-specified attenuation.

Group
Vertical

Syntax

CH<x>:PROBEFunc:EXTAtten <NR3>
CH<x>:PROBEFunc:EXTAtten?

Related Commands

CH<x>:PROBEFunc:EXTDBatten

Arguments

CH<x> is the channel number.

<NR3> is the attenuation value, which is specified as a multiplier in the range from 1.00E-10 to 1.00E+10.

Examples

CH1:PROBEFunc:EXTATTEN 167.00E-3 specifies an external attenuation, which is connected between the your input signal and the input of the probe attached to Channel 1.

CH2:PROBEFunc:EXTATTEN? might return :CH2:PROBEFunc:EXTATTEN 1.0000E+00, indicating that the probe attached to Channel 2 is connected directly to the user's signal.
**CH<x>:PROBEFunc:EXTDBatten**

This command sets or queries the input-output ratio (expressed in decibel units) of external attenuation or gain between the signal and the instrument input channels. The channel is specified by x.

The query form of this command returns the user-specified attenuation in decibels.

**Group**
Vertical

**Syntax**

```
CH<x>:PROBEFunc:EXTDBatten <NR3>
CH<x>:PROBEFunc:EXTDBatten?
```

**Related Commands**

CH<x>:PROBEFunc:EXTAtten

**Arguments**

CH<x> is the channel number.

<NR3> is the attenuation value, which is specified in the range from -200.00 dB to 200.00 dB.

**Examples**

```
CH3:PROBEFUNC:EXTDBATTEN 2.5 specifies an external 2.5 dB attenuator on Channel 3.
CH1:PROBEFUNC:EXTDBATTEN? might return :CH1:PROBEFUNC:EXTDBATTEN 2.5000E+00, indicating that the attenuation for Channel 1 is 2.5 dB.
```

**CH<x>:PROBEFunc:EXTUnits**

This command sets the unit of measurement for the external attenuator of the specified channel. The channel is specified by x. The alternate units are used if they are enabled. Use the CH<x>:PROBEFunc:EXTUnits:STATE command to enable or disable the alternate units.

**Group**
Vertical

**Syntax**

```
CH<x>:PROBEFunc:EXTUnits <QString>
CH<x>:PROBEFunc:EXTUnits?
```

**Related Commands**

CH<x>:PRObe UNIt?
Arguments

CH<x> is the channel number.

<QString> indicates the attenuation unit of measurement for the specified channel.

Examples

CH4:PROBEFunc:EXTUnits "Pascals" sets the unit of measurement for the Channel 4 external attenuator.

CH2:PROBEFunc:EXTUnits? might return :CH2:PROBEFunc:EXTUnits "Pascals", indicating that the Channel 2 external attenuator units of measurement are Pascals.

**CH<x>:PROBEFunc:EXTUnits:STATE**

This command sets or queries the custom units enable state for the specified channel. The channel is specified by x.

Group

Vertical

Syntax

CH<x>:PROBEFunc:EXTUnits:STATE {ON|OFF|<NR1>}

Arguments

CH<x> is the channel number.

OFF argument turns off external units.

ON argument turns on external units.

<NR1> = 0 turns off external units, any other value turns on external units.

Examples

CH2:PROBEFunc:EXTUnits:STATE ON turns on external units.

CH2:PROBEFunc:EXTUnits:STATE? might return :CH2:PROBEFunc:EXTUnits:STATE 0 indicating that external units are off for the specified channel.

**CH<x>:SCAle**

This command sets or returns the vertical scale for the specified analog channel. The channel is specified by x.

Group

Vertical
Syntax

CH<x>:SCAle <NR3>

Arguments

CH<x> is the channel number.

<NR3> is the vertical scale for the specified analog channel.

Examples

CH2:SCALE 200E-3 sets the scale to 200 mV per division.

CH2:SCALE? might return :CH2:SCALE 500.0000E-3 indicating the vertical scale for the specified channel is 500 mV per division.

CH<x>:SCALERATio

This command sets or returns the scale ration for the specified analog channel.

Group

Horizontal

Syntax

CH<x>:SCALERATio <NR2>

CH<x>:SCALERATio?

Arguments

CH<x> is the channel number.

<NR2> is the scale ration for the specified analog channel.

Examples

CH2:SCALERATio 2.0 sets the scale ratio to 2.0.

CH2:SCALERATio? might return :CH2:SCALERATIO 1.0000 indicating the scale ratio is 1.0.

CH<x>:TERmination

This command sets or queries the vertical termination for the specified analog channel. The channel is specified by x.

NOTE. The available arguments depend on the instrument model and the attached accessories.

Group

Vertical
Commands listed in alphabetical order

**Syntax**

CH<x>:TERmination <NR3>
CH<x>:TERmination?

**Arguments**

CH<x> is the channel number.

<NR3> specifies the channel input resistance, which can be specified as 50 Ω or 1,000,000 Ω.

**Examples**

CH4:TERMINATION 50.0E+0 establishes 50 Ω impedance on Channel 1.

CH2:TERMINATION? might return :CH2:TERMINATION 50.0E+0, indicating that Channel 2 is set to 50 Ω impedance.

**CH<x>:VTERm:BIAS**

Sets or queries the termination bias voltage for the specified channel (if control is available).

**Conditions**

Requires a probe with dual inputs.

**Group**

Vertical

**Syntax**

CH<x>:VTERm:BIAS <NR3>
CH<x>:VTERm:BIAS?

**Arguments**

CH<x> is the channel number.

<NR> is the termination voltage.

**Examples**

CH1:VTERm:BIAS 1.5 sets the termination bias voltage on channel 1 to 1.5 Volts

CH2:VTERm:BIAS? might return CH2:VTERm:BIAS 0.0000, indicating that the termination bias voltage for channel 2 is set to 0 volts.

**CH<x>_DALL:LABel:COLor**

This command sets or queries the color of the specified digital group label. The channel is specified by x.

**Group**

Digital
Syntax

**CH<x>_DALL:LABel:COLor <QString>**

Arguments

- **CH<x>** is the channel number.
- **<QString>** is the color of the digital group label. To return the color to the default color, send an empty string as in this example: :CH5_DALL:LABEL:COLOR "".

Examples

- CH1_DALL:LABel:COLor "#FF0000" sets the font color to red.
- CH1_DALL:LABel:COLor? might return :CH1_DALL:LABEL:COLOR "#FFFF00" indicating the font color is yellow.

**CH<x>_DALL:LABel:FONT:BOLD**

This command sets or queries the bold state of the specified digital group. The channel is specified by x.

Group Digital

Syntax

**CH<x>_DALL:LABel:FONT:BOLD {ON|OFF|<NR1>}**

Arguments

- **CH<x>** is the channel number.
- **OFF** argument turns off bold font.
- **ON** argument turns on bold font.
- **<NR1> = 0** turns off bold font; any other value turns on bold font.

Examples

- CH1_DALL:LABel:FONT:BOLD ON sets the font to bold.
- CH1_DALL:LABel:FONT:BOLD? might return :CH1_DALL:LABEL:FONT:BOLD 0 indicating the font is not bold.

**CH<x>_DALL:LABel:FONT:ITALic**

This command sets or queries the italic state of the specified digital group. The channel is specified by x.

Group Digital
Syntax  
CH<x>_DALL:LABel:FONT:ITALic {ON|OFF|<NR1>}

Arguments  
CH<x> is the channel number.
OFF argument turns off italic font.
ON argument turns on italic font.
<NR1> = 0 turns off italic font; any other value turns on italic font.

Examples  
CH1_DALL:LABel:FONT:ITALic 1 turns on italic font.
CH1_DALL:LABel:FONT:ITALic? might return
:CH1_DALL:LABEL:FONT:ITALIC 0 indicating the font is not italic.

**CH<x>_DALL:LABel:FONT:SIZE**

This command sets or queries the font size of the specified digital group. The channel is specified by x.

Group  
Digital

Syntax  
CH<x>_DALL:LABel:FONT:SIZE <NR1>

Arguments  
CH<x> is the channel number.
<NR1> is the font size.

Examples  
CH1_DALL:LABel:FONT:SIZE 16 sets the font size to 16 points.
CH1_DALL:LABel:FONT:SIZE? might return :CH1_DALL:LABEL:FONT:SIZE 20 indicating the font size is 20 points.

**CH<x>_DALL:LABel:FONT:TYPE**

This command sets or queries the font type of the specified digital group, such as Arial or Times New Roman. The channel is specified by x.

Group  
Digital

Syntax  
CH<x>_DALL:LABel:FONT:TYPE <QString>
Arguments

CH<x> is the channel number.

<QString> is the font type.

Examples

CH1_DALL:LABEL:FONT:TYPE "Monosapce" sets the font to a monospace font.


CH<x>_DALL:LABEL:FONT:UNDERline

This command sets or queries the underline state of the specified digital group. The channel is specified by x.

Group

Digital

Syntax

CH<x>_DALL:LABEL:FONT:UNDERline {ON|OFF|<NR1>}

Arguments

CH<x> is the channel number.

OFF argument turns off underline font.

ON argument turns on underline font.

<NR1> = 0 turns off underline font, any other value turns on underline font.

Examples

CH1_DALL:LABEL:FONT:UNDERline ON specifies an underlined font.


CH<x>_DALL:LABEL:NAMe

This command sets or queries the label of the specified digital group. The channel is specified by x.

Group

Digital

Syntax

CH<x>_DALL:LABEL:NAMe <QString>
Arguments  
CH<x> is the channel number.
<br/Qstring> is the name of the group.

Examples  
CH1_DALL:LABEL:NAME "Clock Out" sets the label name to Clock Out.
CH1_DALL:LABEL:NAME? might return:CH1_DALL:LABEL:NAME "This is the digital name".

**CH<x>_D<x>:LABel:COLor**

This command sets or queries the color of the label of the specified digital bit. The channel is specified by x.

Group  
Digital

Syntax  
CH<x>_D<x>:LABel:COLor <QString>

Arguments  
CH<x> is the channel number.
<br/Qstring> is the label color. To return the color to the default color, send an empty string as in this example: :CH5_D1:LABEL:COLOR "".

Examples  
CH1_D1:LABEL:COLOR "#FF0000" sets the color to red.
CH1_D1:LABEL:COLOR? might return:CH1_D1:LABEL:COLOR "#FFFF00" indicating the color is yellow.

**CH<x>_D<x>:LABel:FONT:BOLD**

This command sets or queries the bold state of the label of the specified digital bit. The channel is specified by x.

Group  
Digital

Syntax  
CH<x>_D<x>:LABEL:FONT:BOLD {ON|OFF|<NR1>}

Arguments  
CH<x> is the channel number.
OFF argument turns off bold font.
ON argument turns on bold font.
<NR1> = 0 turns off bold font; any other value turns on bold font.

Examples
CH1_D1:LABEL:FONT:BOLD ON sets the font to bold.
CH1_D1:LABEL:FONT:BOLD? might return :CH1_D1:LABEL:FONT:BOLD 0 indicating the font is not bold.

**CH<x>_D<x>:LABEL:FONT:ITALIC**

This command sets or queries the italic state of the label of the specified digital bit. The channel is specified by x.

**Group** Digital

**Syntax** CH<x>_D<x>:LABEL:FONT:ITALIC {ON|OFF|<NR1>}

**Arguments**
CH<x> is the channel number.
OFF argument turns off italic font.
ON argument turns on italic font.
<NR1> = 0 turns off italic font; any other value turns on italic font.

**Examples**
CH1_D1:LABEL:FONT:ITALIC OFF turns off italic font.

**CH<x>_D<x>:LABEL:FONT:SIZE**

This command sets or queries the font size of the label of the specified digital bit. The channel is specified by x.

**Group** Digital

**Syntax** CH<x>_D<x>:LABEL:FONT:SIZE <NR1>
Arguments  
CH<x> is the channel number.
<NR1> is the font size.

Examples  
CH1_D1:LABel:FONT:SIZE 16 sets the font size to 16 points.
CH1_D1:LABel:FONT:SIZE? might return CH1_D1:LABEL:FONT:SIZE 20 indicating the font size is 20 points.

CH<x>_D<x>:LABEL:FONT:TYPE

This command sets or queries the font type of the label of the specified digital bit, such as Arial or Times New Roman. The channel is specified by x.

Group  
Digital

Syntax  
CH<x>_D<x>:LABEL:FONT:TYPE <QString>

Arguments  
CH<x> is the channel number.
QString is the font type of the label.

Examples  
CH1_D1:LABEL:FONT:TYPE "Monospace" sets the font to Monospace.

CH<x>_D<x>:LABEL:FONT:UNDERline

This command sets or queries the underline state of the label of the specified digital bit. The channel is specified by x.

Group  
Digital

Syntax  
CH<x>_D<x>:LABEL:FONT:UNDERline {ON|OFF|<NR1>}

Arguments  
CH<x> is the channel number.
OFF argument turns off underline font.
ON argument turns on underline font.
<NR1> = 0 turns off underline font, any other value turns on underline font.

Examples

CH1_D1:LABel:FONT:UNDERline ON turns on underline font.
CH1_D1:LABel:FONT:UNDERline? might return
:CH1_D1:LABEL:FONT:UNDERLINE 0 indicating the underline font is off.

**CH<x>_D<x>:LABel:NAME**

Sets or queries the label of the specified digital bit. The channel is specified by x.

**Group**
Digital

**Syntax**

CH<x>_D<x>:LABel:NAME <QString>

**Arguments**

CH<x> is the channel number.

<QString> is the label.

**Examples**

CH1_D1:LABel:NAME "Clock in" sets the name to Clock in.
CH1_D1:LABel:NAME? might return :CH1_D1:LABEL:NAME "Digital 1".

**CLEAR (No Query Form)**

This command clears acquisitions, measurements, and waveforms.

**Group**
Miscellaneous

**Syntax**

CLEAR

**Examples**

CLEAR clears all acquisitions, measurements, and waveforms.
**CLS (No Query Form)**

This command (no query form) clears the following:

- Event Queue
- Standard Event Status Register
- Status Byte Register (except the MAV bit)

If the *CLS command immediately follows an <EOI>, the Output Queue and MAV bit (Status Byte Register bit 4) are also cleared. MAV indicates that information is in the output queue. The device clear (DCL) control message will clear the output queue and thus MAV. *CLS does not clear the output queue or MAV.

*CLS can suppress a Service Request that is to be generated by an *OPC. This will happen if a single sequence acquisition operation is still being processed when the *CLS command is executed.

**Group**  Status and Error

**Syntax**  *CLS

**Related Commands**  DESE  *ESE  *ESR?  EVENT?  EVMsg?  *SRE  *STB?

**Examples**  *CLS clears the instrument status data structures.

**CONFIGuration:ANALOg:BANDWidth? (Query Only)**

This command queries the maximum licensed bandwidth of the instrument.

**Group**  Vertical

**Syntax**  CONFIGuration:ANALOg:BANDwidth?
Returns

The maximum licensed bandwidth of the instrument is returned.

Examples

```
CONFIGuration:ANALog:BANDwidth? might return
:CONFIGURATION:ANALOG:BANDWIDTH 2.0000E+9 indicating the
bandwidth is 2.0 GHz.
```

**CURVe**

This command transfers waveform data to and from the instrument. Each waveform that is transferred has an associated waveform preamble that contains information such as data format and scale.

The CURVe? query transfers data from the instrument. The data source is specified by the DATa SOURce command. The first and last data points are specified by the DATa:STARt and DATa:STOP commands.

The CURVe command transfers waveform data to the instrument. Only one waveform can be transferred at a time. The waveform will only be displayed if the reference is displayed.

For digital sources, CH<x>_D<n> or CH<x>_DALL, when the :DATa:WIDth is 1, the returned data is state only. When the :DATa:WIDth is 2, the returned data is transition data with 2 bits per digital channel representing the transition information as follows:

- **0 0** low
- **0 1** high
- **1 1** multiple transitions in interval ending with high
- **1 0** multiple transitions in interval ending with low

For individual digital channels (such as CH<x>_D<n> ), :DATa:WIDth 2 provides the 2-bit transition data with the upper 14 bits zero. :DATa:WIDth 1 provides only the state in the LSB with the upper 7 bits all zero.

For CH<x>_DAll sources, :DATa:WIDth 2 provides the 2-bit transition data for each of the 8 constituent channels with the D7 bit represented in the 2 most significant bits, D6 in the next 2, and so on. :DATa:WIDth 1 provides the states of each of the 8 constituent channels with D7 represented in the most significant bit and D0 in the least significant bit.

Depending on the sample rate, multi-transition data may not be available and :CURVe? queries for digital channels with :DATa:WIDth 2 may result in a warning event “Execution warning. Multi-transition data not available”. In this case, the transition data returned will be **0 0** or **0 1**.

For MATH sources, only 8-byte double precision floating point data is returned in :CURVe? queries.
Commands listed in alphabetical order

**Group**
Waveform Transfer

**Syntax**

```
CURVe {<Block>|<asc curve>}
CURVe?
```

**Related Commands**
- DATa:SOURce
- DATa:START
- DATa:STOP
- SAVE:WAVEform
- SAVE:WAVEform:FILEFormat
- WFMOutpre?

**Arguments**

- `<Block>` is the waveform data in binary format. The waveform is formatted as:
  
  `#<x><yyy><data><newline>`, where:
  
  - `<x>` is the number of y bytes. For example, if `<yyy>=500`, then `<x>=3`
  
  - `<yyy>` is the number of bytes to transfer. If width is 1, then all bytes on the bus are single data points. If width is 2, then all bytes on the bus are 2-byte pairs. If width is 4, then all bytes on the bus are 4-byte pairs.

- `<data>` is the curve data.

- `<newline>` is a single byte new line character at the end of the data.

**Examples**

```
CURVE <Block>
```
sets the format of the waveform data, transferred to and from the instrument, to binary format.

```
CURVE? with ASCII encoding, start and stop of 1 and 10 respectively, and a width set to 1 might return:
```CURVE 61,62,61,60,60,-59,-59,-58,-58,-58,-59

**NOTE.** `<x>` is hexadecimal format. The letters A-F denote several y bytes between 10 and 15 digits.
NOTE. Curve data is transferred from the instrument asynchronously and, depending upon the length of the curve record, such transfers can require several seconds to complete. During this time, the instrument will not respond to user controls. You can interrupt these asynchronous data transfers by sending a device clear message to the instrument or by interrupting the query with another command or query. Verify that curve data is completely transferred.

It is recommended that you follow such queries with an *ESR? query and verify that the error bit returned and, if set, check the event queue to ascertain the reason for the error. If the error was caused by an interrupted query, then the asynchronous data transfer had not completed when the *ESR? query was sent. In this case, you might need increase your program's time-out value to ensure that all data is transferred and read.

DATa

This command sets or queries the format and location of the waveform data that is transferred with the CURVe command.

Group Waveform Transfer

Syntax DATa {INIT|SNAp}

DATa?

Related Commands CURVe
DATa:STARt
DATa:STOP
DATa:ENCdg
SAVEON:WAVEform FILEFormat
WFMOoutpre:NR_Pt?

Arguments INIT initializes the waveform data parameters to their factory defaults except for DATa:STOP, which is set to the current acquisition record length.

SNAp Sets DATa:STARt and DATa:STOP to match the current waveform cursor positions of WAVEVIEW1 CURSOR1 if these waveform cursors are currently on. If these waveform cursors are not on when the DATa SNAp command is sent, it is silently ignored and DATa:STARt and :STOP remain unchanged.

If either of the cursors is outside the record, DATa:STARt and :STOP are set to nearest values within the record. This will cause DATa:STARt and/or :STOP to snap to the beginning or end of the record whichever is nearest, and the following
Commands listed in alphabetical order

**event is set: 500, "Execution warning", "One or both cursors outside of waveform record".**

If there is no acquired waveform for the cursor source, then DATa:STARt and :STOP remain unchanged and the following event is set: 500, "Execution warning", "No acquired waveform record".

If DATa:STARt is greater than :STOP, the values of DATa:STARt and STOp are swapped and the following event is set: 530, "Data start > stop, Values swapped internally".

**Examples**

DATA INIT initializes the waveform data parameters to their factory defaults.  
DATA? might return DATA:ENCDG RIBINARY;SOURCE CH1;START 1;STOP 1000;FRAMESTART 1;FRAMESTOP 1000.

**DATa:ENCdg**

This command sets or queries the format of outgoing waveform data. This command is equivalent to setting WFMOutpre:ENCdg, WFMOutpre:BN_Fmt, and WFMOutpre:BYT_Or. Setting the DATa:ENCdg value causes the corresponding WFMOutpre values to be updated and vice versa.

**NOTE.** Values are constrained (for outbound data) to the format of the data specified by DATa:SOURce.

**Group**

Waveform Transfer

**Syntax**

DATa:ENCdg {ASCIi|RIBinary|RPBinary|FPBinary|SRIbinary|SRPbinary|SFPbinary}

DATa:ENCdg?

**Related Commands**

WFMOutpre:ENCdg

WFMOutpre:BN_Fmt

WFMOutpre:BYT_Or
Arguments

- **ASCII** specifies the ASCII representation of signed INT, FLOAT. If ASCII is the value, then :BN_FMT and :BYT_OR are ignored.

- **RIBinary** specifies signed integer data point representation with the most significant byte transferred first.
  
  When :BYT_NR is 1, the range is from -128 through 127. When :BYT_NR is 2, the range is from -32,768 through 32,767. When :BYT_NR is 8, then the waveform being queried is set to Fast Acquisition mode. Center screen is 0 (zero). The upper limit is the top of the screen and the lower limit is the bottom of the screen. This is the default argument.

- **RPBinary** specifies the positive integer data point representation, with the most significant byte transferred first.
  
  When :BYT_NR is 1, the range from 0 through 255. When :BYT_NR is 2, the range is from 0 to 65,535. When :BYT_NR is 8, then the waveform being queried is set to Fast Acquisition mode. The center of the screen is 127. The upper limit is the top of the screen and the lower limit is the bottom of the screen.

- **FPBinary** specifies the floating point (width = 4) data.
  
  The range is from $-3.4 \times 10^{38}$ to $3.4 \times 10^{38}$. The center of the screen is 0. The upper limit is the top of the screen and the lower limit is the bottom of the screen.

  The **FPBinary** argument is only applicable to math waveforms or ref waveforms saved from math waveforms.

- **SRIBinary** is the same as RIBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to IBM compatible PCs.

- **SRPBinary** is the same as RPBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.

- **SFPBinary** specifies floating point data in IBM PC format. The SFPBinary argument only works on math waveforms or ref waveforms saved from math waveforms.

Table 2-45: DATa and WFMOutpre Parameter Settings

<table>
<thead>
<tr>
<th>DATa:ENCdg Setting</th>
<th>WFMOutpre Settings</th>
<th>:ENCdg</th>
<th>:BN_FMT</th>
<th>:BYT_OR</th>
<th>:BYT_NR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII</td>
<td>ASCII</td>
<td>N/A</td>
<td>N/A</td>
<td>1,2,4,8</td>
<td></td>
</tr>
<tr>
<td>RIBinary</td>
<td>BIN</td>
<td>RI</td>
<td>MSB</td>
<td>1,2,8</td>
<td></td>
</tr>
<tr>
<td>RPBinary</td>
<td>BIN</td>
<td>RP</td>
<td>MSB</td>
<td>1,2,8</td>
<td></td>
</tr>
<tr>
<td>FPBinary</td>
<td>BIN</td>
<td>FP</td>
<td>MSB</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>SRIBinary</td>
<td>BIN</td>
<td>RI</td>
<td>LSB</td>
<td>1,2,8</td>
<td></td>
</tr>
</tbody>
</table>
Table 2-45: DATa and WFMOutpre Parameter Settings (cont.)

<table>
<thead>
<tr>
<th>DATa:ENCdg Setting</th>
<th>:ENCdg</th>
<th>:BN_FMT</th>
<th>:BYP_OR</th>
<th>:BYP_NR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRPbinary</td>
<td>BIN</td>
<td>RP</td>
<td>LSB</td>
<td>1,2,8</td>
</tr>
<tr>
<td>SFPbinary</td>
<td>BIN</td>
<td>FP</td>
<td>LSB</td>
<td>4</td>
</tr>
</tbody>
</table>

Examples

DATA:ENCdg RPBinary sets the data encoding format to be a positive integer where the most significant byte is transferred first.

DATA:ENCdg? might return :DATa:ENCdg SRPBINARY for the format of the outgoing waveform data.

DATa:SOUrce

This command sets or queries the location of waveform data that is transferred from the instrument by the CURVe? Query.

Group

Waveform Transfer

Syntax

DATa:SOUrce <wfm>[<,><wfm>]
DATa:SOUrce?

Related Commands

CURVe
DATa

Arguments

<wfm> is the location of the waveform data that will be transferred from the instrument to the controller. It can consist of CH<x>, MATH<x>, REF<x>, DIGITALALL. Note that digital data is transferred as 16-bit data, with the least-significant bit representing D0, and the most-significant bit representing D15.

<wfm> can consist of the following:

CH<x> selects the specified analog channel as the source.

MATH<x> selects the specified reference waveform as the source. The reference number is specified by x, which ranges from 1 through 4.

REF<x> selects the specified reference waveform as the source. The reference number is specified by x, which ranges from 1 through 8.

CH<x>_D<x> selects the specified digital channel.

CH<x>_DA11 selects the specified supper channel group of digital channels.
DIGITALALL selects digital waveforms as the source. The Digital data is transferred as 16-bit data, with the least-significant bit representing D0, and the most-significant bit representing D15. The LSB always contains D0-D7 and MSB always contains D8-D15 data.

**Examples**

DATA:SOURCE CH1 specifies that the CH1 waveforms will be transferred in the next CURVe? query.

DATA:SOURCE? might return :DATA:SOURCE REF3, indicating that the source for the waveform data which is transferred using a CURVe? query is reference 3.

**DATa:SOURce:AVAILable? (Query Only)**

This query returns a list of enumerations representing the source waveforms that are currently available for .CURVe? queries. This means that the waveforms have been acquired. If there are none, NONE is returned.

**Group** Waveform Transfer

**Syntax** DATa:SOURce:AVAILable?

**Related Commands** CURVe

**Returns** Returns a list of source waveforms that are currently available for .CURVe? queries.

**Examples** DATA:SOURce:AVAILable? might return :DATA:SOURce:AVAILable CH2, CH3, CH6, CH7, MATH1, REF1 indicating that CH2, CH3, CH6, CH7, MATH1, and REF1 are available.

**DATa:STARt**

This command sets or queries the starting data point for waveform transfer. This command allows for the transfer of partial waveforms to and from the instrument.

**Group** Waveform Transfer

**Syntax** DATa:STARt <NR1>

DATa:STARt?
Related Commands

CURVe
DATa
DATa:STOP
WFMOutpre:NR_Pt?

Arguments

<NR1> is the first data point that will be transferred, which ranges from 1 to the record length. Data will be transferred from <NR1> to DATa:STOP or the record length, whichever is less. If <NR1> is greater than the record length, the last data point in the record is transferred.

DATa:START and DATa:STOP are order independent. When DATa:STOP is greater than DATa:START, the values will be swapped internally for the CURVe? query.

Examples

DATA:START 10 specifies that the waveform transfer will begin with data point 10.

DATA:START? might return :DATA:START 214, indicating that data point 214 is the first waveform data point that will be transferred.

DATa:STOP

This command sets or queries the last data point that will be transferred when using the CURVe? query. This command allows for the transfer of partial waveforms to the controller.

Changes to the record length value are not automatically reflected in the data:stop value. As record length is varied, the DATa:STOP value must be explicitly changed to ensure the entire record is transmitted. In other words, curve results will not automatically and correctly reflect increases in record length if the distance from DATa:START to DATa:STOP stays smaller than the increased record length.

NOTE. When using the CURVe command, DATa:STOP is ignored.

Group

Waveform Transfer

Syntax

DATa:STOP <NR1>
DATa:STOP?

Related Commands

CURVe
DATa
DATa:START
WFMOutpre:NR_Pt?

Arguments

<NR1> is the last data point that will be transferred, which ranges from 1 to the record length. If <NR1> is greater than the record length, then data will be transferred up to the record length. If both DATa:STARt and DATa:STOP are greater than the record length, the last data point in the record is returned.

DATa:STARt and DATa:STOP are order independent. When DATa:STOP is less than DATa:STARt, the values will be swapped internally for the CURVe? query.

If you always want to transfer complete waveforms, set DATa:STARt to 1 and DATa:STOP to the maximum record length, or larger.

Examples

DATA:STOP 15000 specifies that the waveform transfer will stop at data point 15000.

DATA:STOP? might return :DATA:STOP 14900, indicating that 14900 is the last waveform data point that will be transferred.

DATa:WIDth

This command specifies the width, in bytes per point, for waveform data transferred from the oscilloscope via the CURVe? query. (This command is synonymous with WFMOutpre:BYT_Nr.)

Group Waveform Transfer

Syntax

DATa:WIDth <NR1>
DATa:WIDth?

Related Commands WFMOutpre:BYT_Nr

Arguments

<NR1> is an integer that indicates the number of bytes per point for the outgoing waveform data when queried using the CURVe? command. For analog channels the values can be 1 or 2. For digital channels, the values can be 1 or 2. For the digital collection, the values can be 4 or 8.

Examples

DATA:WIDTH 1 sets the data width to 1 byte.

DATA:WIDTH? might return :DATA:WIDTH 1 indicating the width, in bytes per point, for waveform data transferred bya the CURVe? query is 1 byte.
DATE? (Query Only)

This command queries the date that the instrument displays.

Group  
Miscellaneous

Syntax  
DATE?

Related Commands  
TIME?

Returns  
<QString> is a date in the form "yyyy-mm-dd" where yyyy refers to a four-digit year number, mm refers to a two-digit month number from 01 to 12, and dd refers to a two-digit day number in the month.

Examples  

*DDT

This command allows you to specify a command or a list of commands that are executed when the instrument receives a *TRG command. Define Device Trigger (*DDT) is a special alias that the *TRG command uses.

Group  
Miscellaneous

Syntax  
*DDT {<Block>|<QString>}
*DDT?

Related Commands  
ALIas
*TRG

Arguments  
<Block> is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 80 characters. The format of this argument is always returned as a query.

<QString> is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 80 characters.
Examples
d*DDT #217ACQUIRE : STATE  RUN specifies that the acquisition system will be
started each time a *TRG command is sent.

DESE

This command sets and queries the bits in the Device Event Status Enable Register
(DESER). The DESER is the mask that determines whether events are reported to
the Standard Event Status Register (SESR), and entered into the Event Queue.
For a more detailed discussion of the use of these registers, see Registers.

Group
Status and Error

Syntax
DESE <NR1>
DESE?

Related Commands
*CLS
*ESE
*ESR?
EVENT?
EVMsg?
*SRE
*STB?

Arguments
<NR1> The binary bits of the DESER are set according to this value, which ranges
from 1 through 255. For example, DESE 209 sets the DESER to the binary value
11010001 (that is, the most significant bit in the register is set to 1, the next most
significant bit to 1, the next bit to 0, etc.).

The power-on default for DESER is all bits set if *PSC is 1. If *PSC is 0, the
DESER maintains the previous power cycle value through the current power cycle.

NOTE. Setting the DESER and ESER to the same value allows only those codes
to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the
Status Byte Register. Use the *ESE command to set the ESER.

Examples
DESE 209 sets the DESER to binary 11010001, which enables the PON, URQ,
EXE and OPC bits.

DESE? might return :DESE 186, showing that the DESER contains the binary
value 10111010.
Commands listed in alphabetical order

**DIAg:LOOP:OPTion**

This command sets or queries the type of looping desired.

**Group**
Self Test

**Syntax**

```
DIAg:LOOP:OPTion \{FAIL|ONCE|ALWAYS|ONFAIL|NTIMES\}
DIAg:LOOP:OPTion?
```

**Arguments**

- **Fail** - run until a failure is found, then halt.
- **Once** - run through one loop.
- **Always** - run forever.
- **Onfail** - run until a failure is found, then loop on it.
- **Ntimes** - run n number of loops.

**Examples**

```
DIAG:LOOP:OPTION ALWAYS sets loop option to ALWAYS.
```

**DIAg:LOOP:OPTion:NTIMes**

This command sets or queries how many loops to run, if N-times is being used.

**Group**
Self Test

**Syntax**

```
DIAg:LOOP:OPTion:NTIMes <NR1>
DIAg:LOOP:OPTion:NTIMes?
```

**Arguments**

- `<NR1>` is how many loops to run.

**Examples**

```
DIAG:LOOP:OPTION:NTIMES 2 sets diagnostics to loop 2 times.
```

**DIAg:LOOP:STOP (No Query Form)**

Request that diagnostics stop looping.
Commands listed in alphabetical order

**DIAg:LOOP:STOP**

**Syntax**

```
DIAg:LOOP:STOP
```

**Examples**

```
DIAG:LOOP:STOP stops diagnostics looping.
```

**DIAg:MODE**

This command sets or queries the diagnostics mode.

**Syntax**

```
DIAg:MODe {POST|EXTENDED|SERVICE}
DIAg:MODe?
```

**Arguments**

- POST specifies the power on self test diagnostics.
- EXTENDED specifies the extended diagnostics.
- SERVICE specifies the service diagnostics.

**Examples**

```
DIAG:MODE POST specifies the power on self test diagnostics.
```

**DIAg:RESULT? (Query Only)**

This query returns both the overall diagnostics test results and the results of each individual test area.

**Syntax**

```
DIAg:RESULT?
```

**Returns**

The diagnostics results.

**Examples**

```
```
**DIAg:RESUIt:FLAg? (Query Only)**

This query returns the status of the diagnostic test area that has been selected.

**Group**  Self Test

**Syntax**  DIAg:RESUIt:FLAg?

**Returns**  The status of the diagnostics (single area).

**Examples**  DIAg:RESULT:FLAG? might return :DIAG:RESULT:FLAG "NOT RUN" indicating the diagnostics have not been run.

**DIAg:RESUIt:LOG? (Query Only)**

This query returns the test Pass/Fail status of each diagnostic area. It does not return the overall status.

**Group**  Self Test

**Syntax**  DIAg:RESUIt:LOG?

**Returns**  The status of the diagnostic area.

**Examples**  DIAg:RESULT:LOG? might return :DIAG:RESULT:LOG "PASS--IO, PASS--ANALOG, PASS--SYSTEM, PASS--ASIC, PASS--ACQ, PASS--SIGNAL, PASS--MEMORY" indicating the diagnostics result stored in the log.

**DIAg:SELect (No Query Form)**

This command selects or queries an available diagnostic area.

**Group**  Self Test
Commands listed in alphabetical order

**Syntax**

DIAg:SELect {ALL|IO|ANALOG|SYSTEM|ASIC|ACQ|SIGNAL|MEMORY}
DIAg:SELect?

**Arguments**

The argument is the desired diagnostic area.

**Examples**

DIAg:SELECT IO will select the IO group.

**DIAg:STATE (No Query Form)**

This command starts or aborts Self Test. Abort happens after group under test completes.

**Group**

Self Test

**Syntax**

DIAg:STATE {EXECute|ABOrt}

**Arguments**

EXECUTE starts execution of the diagnostics.
ABOrt disables diagnostics capabilities and returns the instrument to a normal operating state.

**Examples**

DIAg:STATE ABORT turns off diagnostics capabilities and returns the instrument to a normal operating state.
DIAg:STATE? might return :DIAg:STATE ABORT, indicating that diagnostics are disabled.

**DIGGRP<x>:D<x>:THReshold**

Sets or queries the threshold level in volts for the specified digital channel. If the source channel doesn't exist, a hardware missing error event is set.

**Group**

Digital

**Syntax**

DIGGRP<x>:D<x>:THReshold <NR3>
DIGGRP<x>:D<x>:THReshold?

**Arguments**

DIGGRP<x> is the channel number.
D<x> is the digital channel bit number (0-7).
<NR3> is the threshold level in volts for the specified digital channel.

**Examples**

DIGGRP1:D1:THRESHold 30.0E-3 sets the threshold to 30 mV.

DIGGRP1:D1:THRESHold? might return :DIGGRP1:D1:THRESHOLD 0.0E+0 indicating the threshold is 0.0 V.

**DISplay? (Query Only)**

This query-only command returns the current Display settings.

**Group** Display Control

**Syntax** DISPlay?

**Examples**

DISPlay? might return :DISPlay:CLOCK 1;COLOR:PALETTE:IMAGEVIEW MONOGREEN; RECORDVIEW TEMPERATURE;USER:GRATICULE 165,50,15;CH1 180,50,100; CH2 300,50,100;CH3 60,50,100;CH4 240,50,100;REF1 0,90,0; REF2 0,90,100;REF3 60,90,100;REF4 240,90,100;MATH1 160,50,100; MATH2 40,60,100;MATH3 120,60,100;MATH4 195,50,100; HISTOGRAM 320,50,100;CARET 150,50,100;MASK 0,25,75;MASKHIGHLIGHT 140,50,100;:DISPlay:COLOR:MATHCOLOR DEFAULT;REFCOLOR DEFAULT; :DISPlay:FILTER SINX;FORMAT YT;GRATICULE IRE;INTENSITY :WAVEFORM:IMAGEVIEW 81.0000;RECORDVIEW 81.0000;:DISPlay :INTENSITY:AUTOBRIGHT 0; :DISPlay:PERSISTENCE OFF;STYLE DOTS;TRIGBAR OFF;TRIGT 1; CURSORTICK LONG;VARPERSIST 2.6000;SCREENTEXT:STATE 1; LABEL1:NAME "";XPOS 100;YP0S 5; :DISPlay:SCREENTEXT :LABEL2:NAME "THIS IS SCREEN TEXT";XPOS 100;YP0S 20; :DISPlay:SCREENTEXT:LABEL3:NAME "";XPOS 100;YP0S 35; :DISPlay:SCREENTEXT:LABEL4:NAME "";XPOS 100;YP0S 50; :DISPlay:SCREENTEXT:LABEL5:NAME "";XPOS 100;YP0S 343; :DISPlay:SCREENTEXT:LABEL6:NAME "";XPOS 100;YP0S 80; :DISPlay:SCREENTEXT:LABEL7:NAME "";XPOS 100;YP0S 95; :DISPlay:SCREENTEXT:LABEL8:NAME "";XPOS 100;YP0S 110; :DISPlay:WAVEFORM 1.

**DISplay:COLors**

Sets or queries the color mode for the graticule and waveform display.
Commands listed in alphabetical order

Group: Display Control

Syntax: DISPLAY:COLORS {NORMAL|INVERTed}
DISPLAY:COLORS?

Arguments:
NORMAL specifies normal color mode.
INVERTed specifies inverted color mode.

Examples:
DISPLAY:COLORS NORMAL sets the display colors to normal.
DISPLAY:COLORS? might return :DISPLAY:COLORS INVERTed indicating
the display colors are inverted.

DISPLAY:GLOBAL:B<x>:STATE

This command sets or queries the global state (display mode On or Off) of the
specified bus. Setting this value true (On or NR1 ≠ 0) turns on the source in the
waveform view. Setting this value false (Off or NR1 = 0) turns off the source
in the waveform view. This command only works if the specified bus is added
already.

Group: Display Control

Syntax: DISPLAY:GLOBAL:B<x>:STATE {<NR1>|OFF|ON}

Arguments:
<NR1> = 0 disables the display of the specified bus; any other value enables
display of the bus.
ON enables display of the specified bus.
OFF disables display of the specified bus.

Examples:
DISPLAY:GLOBAL:B1:STATE OFF turns off the display of bus 1.
indicating the bus is displayed.

DISPLAY:GLOBAL:CH<x>:STATE

This command sets or queries the global state (display mode On or Off) of the
specified channel (both analog and digital). Setting this value true (On or NR1
≠ 0 ) turns on the source in the waveform view. Setting this value false (Off or 
NR1 = 0 ) turns off the source in the waveform view. This command only works 
if the specified channel is added already.

Group Display Control

Syntax DISplay:GLObal:CH<x>:STATE {<NR1>|OFF|ON}

Arguments <NR1> = 0 disables the display of the specified channel; any other value enables 
display of the channel.
ON enables display of the specified channel.
OFF disables display of the specified channel.

Examples DISplay:GLObal:CH1:STATE OFF turns off the display of channel 1.
0 indicating that channel 1 is not displayed.

DI$play:GLObal:MATH<x>:STATE

This command sets or queries the global state (display mode On or Off) of the 
specified math. Setting this value true (On or NR1 ≠ 0 ) turns on the source in the 
waveform view. Setting this value false (Off or NR1 = 0 ) turns off the source in 
the waveform view. This command only works if the specified math waveform 
is added already.

Group Display Control

Syntax DISplay:GLObal:MATH<x>:STATE {<NR1>|OFF|ON}

Arguments <NR1> = 0 disables the display of the specified math, any other value enables 
display of the math.
ON enables display of the specified math.
OFF disables display of the specified math.

Examples DISplay:GLObal:MATH1:STATE 1 displays math 1.
DISplay:GLObal:MATH1:STATE? might return :DISPLAY:GLOBAL:MATH1:STATE 
1 indicating that math1 is displayed.
**DISplay:GLObal:PLOT<x>:STATE**

This command sets or queries the global state (display mode On or Off) of the specified time trend plot. Setting this value true (On or NR1 ≠ 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified plot is added already.

**Group**
Display Control

**Syntax**
DISplay:GLObal:PLOT<x>:STATE \{<NR1>|OFF|ON\}

**Arguments**

- PLOT<x> is the plot number.
- <NR1> = 0 disables the display of the specified plot, any other value enables display of the plot.
- ON enables display of the specified plot.
- OFF disables display of the specified plot.

**Examples**

DISplay:GLObal:PLOT1:STATE ON displays plot 1.

**DISplay:GLObal:REF<x>:STATE**

This command sets or queries the global state (display mode On or Off) of the specified reference waveform. Setting this value true (On or NR1 ≠ 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified reference waveform is added already.

**Group**
Display Control

**Syntax**
DISplay:GLObal:REF<x>:STATE \{<NR1>|OFF|ON\}

**Arguments**

- REF<x> is the Reference waveform number.
- <NR1> = 0 disables the display of the specified reference, any other value enables display of the reference.
- ON enables display of the specified reference.
OFF disables display of the specified reference.

**Examples**

```
DISPLAY:GLOBAL:REF1:STATE? might return
:DISPLAY:GLOBAL:REF1:STATE 1 indicating that reference 1 is displayed.
```

**DI-Splay:INTENSITY? (Query Only)**

This query-only command returns the waveform saturation level and screen saver settings.

**Group** Display Control

**Syntax** `DISPLAY:INTENSITY?`

**Related Commands**

**Arguments** None

**Examples**

```
DISPLAY:INTENSITY? might return :DISPLAY:INTENSITY:BACKLIGHT 1, indicating the intensity is set to 1.
```

**DI-Splay:INTENSITY:BACKLight**

This command sets or queries the display backlight intensity setting.

**Group** Display Control

**Syntax**

```
DISPLAY:INTENSITY:BACKLight {LOW|MEDIUM|HIGH}
DISPLAY:INTENSITY:BACKLight?
```

**Arguments**

- **LOW** selects a low brightness level.
- **MEDIUM** selects a moderate brightness level.
- **HIGH** selects a full brightness level.

**Examples**

```
DISPLAY:INTENSITY:BACKLIGHT LOW sets the display backlight to low brightness level.
```
DISPLAY:INTENSITY:BACKLIGHT? might return 
DISPLAY:INTENSITY:BACKLIGHT HIGH, indicating that the display backlight 
is set to full brightness level.

**DISPLAY:INTENSITY:BACKLight:AUTODim:ENaBLE**

Sets or queries the state of the display auto-dim feature. The default is enabled. 
Once the backlight has dimmed, any button push, knob turn or mouse movement 
returns the backlight value to the value set by :DISPLAY:INTENSITY:BACKLight.

<table>
<thead>
<tr>
<th>Group</th>
<th>Display Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>DISPLAY:INTENSITY:BACKLight:AUTODim:ENABLE {ON</td>
</tr>
<tr>
<td>Related Commands</td>
<td>DISPLAY:INTENSITY:BACKLight</td>
</tr>
<tr>
<td>Arguments</td>
<td>ON enables the display auto-dim feature.</td>
</tr>
<tr>
<td></td>
<td>OFF disables the display auto-dim feature.</td>
</tr>
<tr>
<td>Examples</td>
<td>DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE ON enables</td>
</tr>
<tr>
<td></td>
<td>auto-dimming of the backlight.</td>
</tr>
<tr>
<td></td>
<td>DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE ON, indicating that</td>
</tr>
<tr>
<td></td>
<td>auto-dimming of the backlight is enabled.</td>
</tr>
</tbody>
</table>

**DISPLAY:INTENSITY:BACKLight:AUTODim:TIMe**

Sets or queries the amount of time, in minutes, to wait for no user interface activity 
before automatically dimming the display. The time can range from a minimum of 
10 minutes to a maximum of 1440 minutes (24 hours). The default is 10 minutes.

<table>
<thead>
<tr>
<th>Group</th>
<th>Display Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>DISPLAY:INTENSITY:BACKLight:AUTODim:TIME &lt;NR1&gt;</td>
</tr>
<tr>
<td></td>
<td>DISPLAY:INTENSITY:BACKLight:AUTODim:TIME?</td>
</tr>
<tr>
<td>Arguments</td>
<td>&lt;NR1&gt; is the amount of time, in minutes, to wait for no user interface activity</td>
</tr>
<tr>
<td></td>
<td>before automatically dimming the display.</td>
</tr>
</tbody>
</table>
Examples

DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:TIME 30 sets the backlight to autodim in 30 minutes.


DISPLAY:MATHFFTView<x>:AUTOScale

This command sets or returns the enabled state of autoscale for Math/FFT waveforms.

Group

Display Control

Syntax

DISPLAY:MATHFFTView<x>:AUTOScale {OFF|ON|<NR1>

DISPLAY:MATHFFTView<x>:AUTOScale?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

OFF disables the autoscale feature.

ON enables the autoscale feature.

<NR1> = 0 disables the autoscale feature; any other value enables the autoscale feature.

Examples

DISPLAY:MATHFFTView1:AUTOScale OFF disables the autoscale feature.

DISPLAY:MATHFFTView3:AUTOScale? might return DISPLAY:MATHFFTVIEW3:AUTOSCALE 1, indicating that autoscale is on for the Math3 FFT waveform.

DISPLAY:MATHFFTView<x>:CURSor:ASOUrce? (Query Only)

This command queries the Math-FFT waveform view source for cursor A.

Group

Cursor

Syntax

DISPLAY:MATHFFTView<x>:CURSor:ASOURce?

Arguments

MATHFFTView<x> is the Math-FFT waveform number.
Examples

DISPLAY:MATHFFTView1:CURSOR:ASOURCE? might return
DISPLAY:MATHFFTVIEW1:CURSOR:ASOURCE MATH1, indicating the source is MATH 1.

DISPLAY:MATHFFTView<x>:CURSOR:BSOURCE? (Query Only)

This command queries the Math-FFT waveform view source for cursor B.

Group Cursor

Syntax DISPLAY:MATHFFTView<x>:CURSOR:BSOURCE?

Arguments MATHFFTView<x> is the Math-FFT waveform number.

Examples DISPLAY:MATHFFTView5:CURSOR:BSOURCE? might return
DISPLAY:MATHFFTVIEW5:CURSOR:BSOURCE MATH5, indicating the source is Math waveform MATH 5.

DISPLAY:MATHFFTView<x>:CURSOR:DDT? (Query Only)

This command queries the delta Y over delta X (ΔY/ΔX) cursor readout value of the specified cursor in the specified Math-FFT view.

NOTE. If the current cursor is set to horizontal mode, this command will time out.

Group Cursor

Syntax DISPLAY:MATHFFTView<x>:CURSOR:DDT?

Arguments MATHFFTView<x> is the Math-FFT waveform number.

Examples DISPLAY:MATHFFTView<x>:CURSOR:DDT? might return
**DISplay:MATHFFTView<x>:CURSor:FUNCTION**

This command sets or queries the cursor type for the specified Math-FFT view.

**Group**

Cursor

**Syntax**

DISplay:MATHFFTView<x>:CURSor:FUNCTION
{WAVEform|VBArS|HBArS|SCREEN}

DISplay:MATHFFTView<x>:CURSor:FUNCTION?

**Arguments**

MATHFFTView<x> is the Math-FFT waveform number.

WAVEFORM specifies to display the paired vertical cursors in YT display format for measuring waveform amplitude and time. Measurements are taken at where the cursor intersects the waveform, and tracks waveform changes.

VBArS specifies vertical bar cursors, which measure in horizontal units.

HBArS specifies horizontal bar cursors, which measure in vertical units.

SCREEN specifies to display both horizontal and vertical bar cursors, which display the horizontal and vertical positions of the cursors, not waveform levels. Use these cursors to measure anywhere in the waveform display area.

**Examples**

DISplay:MATHFFTView5:CURSor:FUNCTION HBArS sets Math-FFT waveform 5 to use horizontal bar cursors.

DISplay:MATHFFTView1:CURSor:FUNCTION? might return DISplay:MATHFFTView1:CURSor:FUNCTION? SCREEN, indicating that Math-FFT waveform 2 is set to use both horizontal and vertical cursors.

**DISplay:MATHFFTView<x>:CURSor:HBArS:APOsition**

This command sets or returns the position of horizontal cursor A for the specified Math-FFT view.

**Group**

Cursor

**Syntax**

DISplay:MATHFFTView<x>:CURSor:HBArS:APOsition <NR3>

DISplay:MATHFFTView<x>:CURSor:HBArS:APOsition?

**Arguments**

MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the cursor position of the specified cursor in the specified view.
Examples

```
DISPLAY:MATHFFTView1:CURSOR:HBARS:APOSITION 50.0e-3 sets the position to 50 mV.
```

```
```

**DISPLAY:MATHFFTView<x>:CURSOR:HBARS:AUNIts? (Query Only)**

This command queries the vertical units of horizontal cursor A for the specified Math-FFT view.

**Group**

Cursor

**Syntax**

```
DISPLAY:MATHFFTView<x>:CURSOR:HBARS:AUNIts?
```

**Arguments**

`MATHFFTView<x>` is the Math-FFT waveform number.

**Examples**

```
DISPLAY:MATHFFTView1:CURSOR:HBARS:AUNIts? might return DISPLAY:MATHFFTView1:CURSOR:HBARS:AUNITS "dBm", indicating the vertical units are dBm.
```  

**DISPLAY:MATHFFTView<x>:CURSOR:HBARS:BPOSITION**

This command sets or returns the position of horizontal cursor B for the specified Math-FFT view.

**Group**

Cursor

**Syntax**

```
DISPLAY:MATHFFTView<x>:CURSOR:HBARS:BPOSITION <NR3>
DISPLAY:MATHFFTView<x>:CURSOR:HBARS:BPOSITION?
```

**Arguments**

`MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` is the vertical cursor B position for the specified Math-FFT view.

**Examples**

```
DISPLAY:MATHFFTView1:CURSOR:HBARS:BPOSITION 50e-3 sets the position to 0.005 vertical units.
```
Commands listed in alphabetical order

**DISPLAY:MATHFFTVIEW1:CURSOR:HBARS:BPOSITION?** might return
DISPLAY:MATHFFTVIEW1:CURSOR:HBARS:BPOSITION -40.4061, indicating
the position is -40.4061 vertical units.

**DISPLAY:MATHFFTVIEW<x>:CURSor:HBArS:BUNIts? (Query Only)**

This command queries the vertical units of horizontal cursor B for the specified
Math-FFT view.

**Group**
Cursor

**Syntax**
DISPLAY:MATHFFTVIEW<x>:CURSOR:HBARS:BUNIts?

**Arguments**
MATHFFTVIEW<x> is the Math-FFT waveform number.

**Examples**
DISPLAY:MATHFFTVIEW3:CURSOR:HBARS:BUNITS "dBm", indicating the
cursor units are dBm.

**DISPLAY:MATHFFTVIEW<x>:CURSor:HBArS:DELTa? (Query Only)**

This command queries the horizontal cursor’s delta value of the specified
Math-FFT view.

**Group**
Cursor

**Syntax**
DISPLAY:MATHFFTVIEW<x>:CURSOR:HBARS:DELTa?

**Arguments**
MATHFFTVIEW<x> is the Math-FFT waveform number.

**Examples**
DISPLAY:MATHFFTVIEW1:CURSOR:HBARS:DELTa? might return
DISPLAY:MATHFFTVIEW1:CURSOR:HBARS:DELTa 30.61448, indicating the
cursor readout is 30.61448 units.

**DISPLAY:MATHFFTVIEW<x>:CURSor:MODe**

This command sets or queries the cursor tracking mode of the specified Math-FFT
view.
Commands listed in alphabetical order

**Group** Cursor

**Syntax**

```
DISplay:MATHFFTView<x>:CURSor:MODE {INDEPENDENT|TRACK}
```

**Arguments**

- `MATHFFTView<x>` is the Math-FFT waveform number.
- INDEPENDENT allows independent adjustment of the two cursors.
- TRACK ties the navigational functionality of the two cursors together. For cursor 1 adjustments, this ties the movement of the two cursors together, however, cursor 2 continues to move independently of cursor 1.

**Examples**

```
DISplay:MATHFFTView1:CURSor:MODE INDEPENDENT allows independent adjustment of the two cursors.
DISplay:MATHFFTView1:CURSor:MODE? might return
:DISPLAY:MATHFFTVIEW1:CURSOR:MODE INDEPENDENT, indicating the mode is set to independent.
```

**DISplay:MATHFFTView<x>:CURSor:ONEOVERDELTATVALUE? (Query Only)**

This command queries the one over delta cursor readout value of the specified Math-FFT view.

**Group** Cursor

**Syntax**

```
DISplay:MATHFFTView<x>:CURSor:ONEOVERDELTATVALUE?
```

**Arguments**

- `MATHFFTView<x>` is the Math-FFT waveform number.

**Examples**

```
DISplay:MATHFFTView1:CURSor:ONEOVERDELTATVALUE? might return
DISPLAY:MATHFFTVIEW1:CURSOR:ONEOVERDELTATVALUE 2.6005E-9, indicating the one over delta time value is 2.6005 nanoseconds.
```

**DISplay:MATHFFTView<x>:CURSor:SCREEN:AXPOSition**

This command sets or returns the vertical cursor A x-axis waveform measurement position of the specified Math-FFT view.

**Group** Cursor
Commands listed in alphabetical order

[Syntax]

```
DISPLAY:MATHFFTVIEW<x>:CURSOR:SCREEN:AXPOSITION <NR3>
DISPLAY:MATHFFTVIEW<x>:CURSOR:SCREEN:AXPOSITION?
```

[Arguments]

- `MATHFFTVIEW<x>` is the Math-FFT waveform number.
- `<NR3>` is the cursor position in MHz.

[Examples]

```
DISPLAY:MATHFFTVIEW1:CURSOR:SCREEN:AXPOSITION 700.000E+6 sets the cursor position to 700 MHz.
DISPLAY:MATHFFTVIEW3:CURSOR:SCREEN:AXPOSITION 125.000E+6, indicating the x-axis value of cursor A is 125 MHz.
```

**DISplay:MATHFFTVIEW<x>:CURSor:SCREEN:AYPOSition**

This command sets or returns the vertical cursor A y-axis amplitude measurement value of the specified Math-FFT view.

[Group]

Cursor

[Syntax]

```
DISPLAY:MATHFFTVIEW<x>:CURSOR:SCREEN:AYPOSITION <NR3>
DISPLAY:MATHFFTVIEW<x>:CURSOR:SCREEN:AYPOSITION?
```

[Arguments]

- `MATHFFTVIEW<x>` is the Math-FFT waveform number.
- `<NR3>` is the cursor A position of the specified cursor in the specified view.

[Examples]

```
DISPLAY:MATHFFTVIEW2:CURSOR:SCREEN:AYPOSITION 1.0e0 sets the cursor position to 1.0 dBm.
DISPLAY:MATHFFTVIEW5:CURSOR:SCREEN:AYPOSITION? might return DISPLAY:MATHFFTVIEW5:CURSOR:SCREEN:AYPOSITION 1.1741714106633, indicating the cursor position is 1.174 vertical units (such as dBm).
```

**DISplay:MATHFFTVIEW<x>:CURSor:SCREEN:BXPOSition**

This command sets or returns the vertical cursor Bx-axis waveform time measurement position of the specified Math-FFT view.

[Group]

Cursor
### Command Overview

**Display:MATHFFTView<x>:CURSor:SCREEN:BXPOSition**

**Syntax**

```
DISPLAY:MATHFFTView<x>:CURSor:SCREEN:BXPOSition <NR3>
DISPLAY:MATHFFTView<x>:CURSor:SCREEN:BXPOSition?
```

**Arguments**

- `MATHFFTView<x>` is the Math-FFT waveform number.
- `<NR3>` is the horizontal cursor B position of the specified cursor in the specified view.

**Examples**

- `DISPLAY:MATHFFTView1:CURSor:SCREEN:BXPOSition 3.5e9` sets the cursor position to 3.5 GHz.
- `DISPLAY:MATHFFTView1:CURSor:SCREEN:BXPOSition?` might return `DISPLAY:MATHFFTView1:CURSOR:SCREEN:BXPOSITION 2.50E+9`, indicating the cursor position is 2.5 GHz.

**Display:MATHFFTView<x>:CURSor:SCREEN:BYPOSition**

This command sets or returns the vertical cursor B y-axis amplitude measurement value of the specified Math-FFT view.

**Group**

Cursor

**Syntax**

```
DISPLAY:MATHFFTView<x>:CURSor:SCREEN:BYPOSition <NR3>
DISPLAY:MATHFFTView<x>:CURSor:SCREEN:BYPOSition?
```

**Arguments**

- `MATHFFTView<x>` is the Math-FFT waveform number.
- `<NR3>` is the vertical cursor B position of the specified cursor in the specified view.

**Examples**

- `DISPLAY:MATHFFTView1:CURSor:SCREEN:BYPOSition -50.0e0` sets the cursor position to -50.0.

**Display:MATHFFTView<x>:CURSor:STATE**

This command sets or queries the visible state of cursors for the specified Math-FFT view.

**Group**

Cursor

---

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer 2-279
Commands listed in alphabetical order

Syntax

`DISPLAY:MATHFFTVIEW<x>:CURSoR:STATE {ON|OFF|<NR1>}`

`DISPLAY:MATHFFTVIEW<x>:CURSoR:STATE?`

Arguments

`MATHFFTVIEW<x>` is the Math-FFT waveform number.

1 or ON enables the cursors.

0 or OFF disables the cursors.

`<NR1>` = 0 turns off cursors; any other value displays cursors.

Examples

`DISPLAY:MATHFFTVIEW2:CURSoR:STATE OFF` turns off the cursors for Math-FFT waveform 2.

`DISPLAY:MATHFFTVIEW1:CURSoR:STATE?` might return `DISPLAY:MATHFFTVIEW1:CURSoR:STATE 1` indicating the cursors are enabled.

`DISPLAY:MATHFFTVIEW<x>:CURSoR:VBARS:APOSition`:

This command sets or queries the horizontal cursor A position for the specified Math-FFT view

Group Cursor

Syntax

`DISPLAY:MATHFFTVIEW<x>:CURSoR:VBARS:APOSition <NR3>`

`DISPLAY:MATHFFTVIEW<x>:CURSoR:VBARS:APOSition?`

Arguments

`MATHFFTVIEW<x>` is the Math-FFT waveform number.

`<NR3>` sets the vertical cursor A position in the specified view.

Examples

`DISPLAY:MATHFFTVIEW1:CURSoR:VBARS:APOSition 500.0e6` sets the cursor A position to 500 MHz.


`DISPLAY:MATHFFTVIEW<x>:CURSoR:VBARS:BPOSition`:

This command sets or queries the vertical cursor B position for the specified Math-FFT view.
Group  Cursor
Syntax  DISplay:MATHFFTView<x>:CURSor:VBArS:BPOSition <NR3>
       DISplay:MATHFFTView<x>:CURSor:VBArS:BPOSition?
Arguments  MATHFFTView<x> is the Math-FFT waveform number.
           <NR3> sets the vertical cursor B position in the specified view.
Examples  DISplay:MATHFFTView1:CURSor:VBArS:BPOSition 2.0e9 sets the cursor
          B position to 2.0 GHz.
          DISplay:MATHFFTView1:CURSor:VBArS:BPOSition? might return
          DISPLAY:MATHFFTVIEW1:CURSOR:VBARS:BPOSITION 2.50E+9, indicating
          the cursor position is at 2.5 GHz.

DISplay:MATHFFTView<x>:CURSor:VBArS:DELTa? (Query Only)
This command queries the vertical cursor's delta T readout value for the specified
Math-FFT view.
Group  Cursor
Syntax  DISplay:MATHFFTView<x>:CURSor:VBArS:DELTa?
Arguments  MATHFFTView<x> is the Math-FFT waveform number.
Examples  DISplay:MATHFFTView1:CURSor:VBArS:DELTa? might return
          DISPLAY:MATHFFTVIEW1:CURSOR:VBARS:DELTA 1.9144661308840E+9,
          indicating the delta T cursor value is 1.9 GHz.

DISplay:MATHFFTView<x>:CURSor:VBArS:AUNIts? (Query Only)
This command queries the vertical cursor A measurement units for the specified
Math-FFT view.
Group  Cursor
Syntax  DISplay:MATHFFTView<x>:CURSor:VBArS:AUNIts?
Arguments \[
\text{MATHFFTView}<x> \text{ is the Math-FFT waveform number.}
\]

Examples \[
\text{DISplay:MATHFFTView}<1>:\text{CURSor:VBAr}:\text{UNIts?} \text{ might return}
\Rightarrow \text{DISplay:MATHFFTView}<1>:\text{CURSor:VBAr}:\text{UNIts} \text{ "dBm"}, \text{indicating the}
\text{cursor units are dBm.}
\]

**DISplay:MATHFFTView<x>:CURSor:VBAr:s:BUNIts? (Query Only)**

This command queries the vertical cursor B measurement units for the specified Math-FFT view.

Group Cursor

Syntax \[
\text{DISplay:MATHFFTView}<x>:\text{CURSor:VBAr:s:BUNIts?}
\]

Arguments \[
\text{MATHFFTView}<x> \text{ is the Math-FFT waveform number.}
\]

Examples \[
\text{DISplay:MATHFFTView}<1>:\text{CURSor:VBAr:s:BUNIts?} \text{ might return}
\Rightarrow \text{DISplay:MATHFFTView}<1>:\text{CURSor:VBAr:s:BUNIts} \text{ "Hz" indicating the}
\text{cursor units are Hz.}
\]

**DISplay:MATHFFTView<x>:CURSor:WAVEform:APOSition**

This command sets or queries the waveform cursor A position for the specified Math-FFT view.

Group Cursor

Syntax \[
\text{DISplay:MATHFFTView}<x>:\text{CURSor:WAVEform:APOSition} <NR3>
\text{DISplay:MATHFFTView}<x>:\text{CURSor:WAVEform:APOSition?}
\]

Arguments \[
\text{MATHFFTView}<x> \text{ is the Math-FFT waveform number.}
\text{<NR3> is the waveform cursor A position in the specified plot view}
\]

Examples \[
\text{DISplay:MATHFFTView}<1>:\text{CURSor:WAVEform:APOSition} 600.0e6 \text{ sets the}
\text{position to 600 MHz.}
\]
DISPLAY:MATHFFTView1:CURSor:WAVEform:APOSITION? might return
DISPLAY:MATHFFTView1:CURSor:WAVEform:APOSITION 25.000E+6,
indicating the cursor position is 25 MHz.

DISPLAY:MATHFFTView<x>:CURSor:WAVEform:BPOSITION

This command sets or queries the waveform cursor B position for the specified Math-FFT view.

**Group**  Cursor

**Syntax**  
- DISPLAY:MATHFFTView<x>:CURSor:WAVEform:BPOSITION <NR3>
- DISPLAY:MATHFFTView<x>:CURSor:WAVEform:BPOSITION?

**Arguments**  
- MATHFFTView<x> is the Math-FFT waveform number.
- <NR3> is the waveform cursor B position in the specified plot view.

**Examples**  
- DISPLAY:MATHFFTView1:CURSor:WAVEform:BPOSITION 2.0E+9 sets the cursor position to 2.0 GHz.
- DISPLAY:MATHFFTView1:CURSor:WAVEform:BPOSITION? might return
  DISPLAY:MATHFFTView1:CURSor:WAVEform:BPOSITION 250.0E+6,
  indicating the waveform cursor B position is 250 MHz.

DISPLAY:MATHFFTView<x>:GRIDlines

This command sets or queries the grid lines setting for the specified Math-FFT view.

**Group**  Display Control

**Syntax**  
- DISPLAY:MATHFFTView<x>:GRIDlines {HORizontal|VERTical|BOTH}
- DISPLAY:MATHFFTView<x>:GRIDlines?

**Arguments**  
- MATHFFTView<x> is the Math-FFT waveform number.
- HORizontal specifies horizontal grid lines.
- VERTical specifies vertical grid lines.
- BOTH specifies both vertical and horizontal grid lines.
Examples

```
DISplay:MATHFFTView1:GRIDlines VERTICAL specifies vertical grid lines.
DISplay:MATHFFTView1:GRIDlines? might return
:DISPLAY:MATHFFTVIEW1:GRIDLINES BOTH indicating both horizontal and vertical grid lines are displayed.
```

**DISplay:MATHFFTView<x>:MATH:MATH<x>:STATE**

This command sets or queries the display state of the specified math waveform for the specified Math-FFT view.

**Group**
Display Control

**Syntax**

```
DISplay:MATHFFTView<x>:MATH:MATH<x>:STATE {OFF|ON|<NR1>}
```

**Arguments**

- `MATHFFTView<x>` is the Math-FFT waveform number.
- `OFF` disables displaying the specified Math-FFT view.
- `ON` enables displaying the specified Math-FFT view.
- `<NR1> = 0` disables the specified Math-FFT view, any other value enables the specified Math-FFT view.

**Examples**

```
DISplay:MATHFFTView1:MATH:MATH1:STATE ON enables the specified Math-FFT view
DISplay:MATHFFTView4:MATH:MATH4:STATE? might return
:DISPLAY:MATHFFTVIEW1:MATH:MATH1:STATE 1, indicating the waveform Math-FFT 1 is displayed.
```

**DISplay:MATHFFTView<x>:XAXIS:SCALE**

This command sets or queries the x-axis scale (Linear or Log) for the specified Math-FFT view.

**Group**
Display Control

**Syntax**

```
DISplay:MATHFFTView<x>:XAXIS:SCALE {LINEAr|LOG}
DISplay:MATHFFTView<x>:XAXIS:SCALE?
```
**Arguments**

- **MATHFFTView\(<x>\)** is the Math-FFT waveform number.
  - **LINEAR** specifies a linear scale.
  - **LOG** specifies a logarithmic scale.

**Examples**

- `DISplay:MATHFFTView1:XAXIS:SCALE LOG` specifies a logarithmic scale.
- `DISplay:MATHFFTView1:XAXIS:SCALE?` might return `DISPLAY:MATHFFTVIEW1:XAXIS:SCALE LINEAR`, indicating that the scale setting is linear.

**DISplay:MATHFFTView\(<x>\):YAXIS:SCALE**

This command sets or queries the vertical scale setting (Linear or dBm) for the specified Math-FFT view.

**Group**

Display Control

**Syntax**

```
DISplay:MATHFFTView\(<x>\):YAXIS:SCALE {LINEAR|DBM}
DISplay:MATHFFTView\(<x>\):YAXIS:SCALE?
```

**Arguments**

- **MATHFFTView\(<x>\)** is the Math-FFT waveform number.
  - **LINEAR** specifies a linear scale.
  - **DBM** specifies a dBm scale.

**Examples**

- `DISplay:MATHFFTView1:YAXIS:SCALE LINEAR` specifies a linear scale.
- `DISplay:MATHFFTView1:YAXIS:SCALE?` might return `DISPLAY:MATHFFTVIEW1:YAXIS:SCALE DBM`, indicating that the scale is dBm.

**DISplay:MATHFFTView\(<x>\):ZOOM:XAXIS:FROM**

This command sets or queries the value of the left edge of the zoom area for the specified Math-FFT view.

**Group**

Zoom

**Syntax**

```
DISplay:MATHFFTView\(<x>\):ZOOM:XAXIS:FROM <NR3>
DISplay:MATHFFTView\(<x>\):ZOOM:XAXIS:FROM?
```
Arguments

- **MATHFFTView<x>** is the Math-FFT waveform number.
- **<NR3>** is the value of the left edge of the zoom x axis in the specified plot view.

Examples

- `DISPLAY:MATHFFTView1:ZOOM:XAXIS:FROM 800.e6` sets the x axis value of the left edge of the plot view to 800 MHz.
- `DISPLAY:MATHFFTView1:ZOOM:XAXIS:FROM?` might return `DISPLAY:MATHFFTVIEW1:ZOOM:XAXIS:FROM 781.0E+6`, indicating the x axis value of the left edge of the plot view is at 781 MHz.

**DISPLAY:MATHFFTView<x>:ZOOM:XAXIS:TO**

This command sets or queries the value of the right edge value of the zoom area for the specified Math-FFT view.

Group

- **Zoom**

Syntax

- `DISPLAY:MATHFFTView<x>:ZOOM:XAXIS:TO <NR3>`

Arguments

- **MATHFFTView<x>** is the Math-FFT waveform number.
- **<NR3>** is the value of the right edge of the zoom x axis in the specified plot view.

Examples

- `DISPLAY:MATHFFTView1:ZOOM:XAXIS:TO 2.3E+9` sets the value of the right edge of the x axis to 2.3 GHz.
- `DISPLAY:MATHFFTView1:ZOOM:XAXIS:TO?` might return `DISPLAY:MATHFFTVIEW1:ZOOM:XAXIS:TO 2.343750E+9`, indicating the value of the right edge of the zoom x axis is 2.34 GHz.

**DISPLAY:MATHFFTView<x>:ZOOM:YAXIS:FROM**

This command sets or queries the bottom edge value of the zoom y-axis area for the specified Math-FFT view.

Group

- **Zoom**

Syntax

- `DISPLAY:MATHFFTView<x>:ZOOM:YAXIS:FROM <NR3>`
- `DISPLAY:MATHFFTView<x>:ZOOM:YAXIS:FROM?`
Arguments
\text{MATHFFTView}^{<x>\text{ is the Math-FFT waveform number.}}

\text{<NR3\text{ is the bottom value of the zoom y axis in the specified plot view.}}

Examples
\text{DISPLAY:MATHFFTView1:ZOOM:YAXIS:FROM -75.0e0 sets the bottom value of the y axis to -75 dBm.}

\text{DISPLAY:MATHFFTView1:ZOOM:YAXIS:FROM? might return}
\text{DISPLAY:MATHFFTVIEW1:ZOOM:YAXIS:FROM -78.8258285893367 indicating the bottom value of the y axis is -78.8 dBm.}

\text{DISPLAY:MATHFFTView}^{<x>\text{:ZOOM:YAXIS:TO}}

This command sets or queries the top edge value of the zoom y-axis area for the specified Math-FFT view.

Group
Zoom

Syntax
\text{DISPLAY:MATHFFTView}^{<x>\text{:ZOOM:YAXIS:TO <NR3>}}

\text{DISPLAY:MATHFFTView}^{<x>\text{:ZOOM:YAXIS:TO?}}

Arguments
\text{MATHFFTView}^{<x>\text{ is the Math-FFT waveform number.}}

\text{<NR3\text{ is the top value of the zoom y axis in the specified plot view.}}

Examples
\text{DISPLAY:MATHFFTView1:ZOOM:YAXIS:TO 21.0E0 sets the top value of the zoom y axis in the specified plot view to 21 dBm.}

\text{DISPLAY:MATHFFTView1:ZOOM:YAXIS:TO? might return}
\text{DISPLAY:MATHFFTVIEW1:ZOOM:YAXIS:TO 21.1741714106633 indicating the top value of the zoom y axis in the specified plot view is 21.17 dBm.}

\text{DISPLAY:PERSistence}

This command sets or queries the display persistence for analog waveforms. Persistence is valid for wave views only.

Group
Display Control

Syntax
\text{DISPLAY:PERSistence \{OFF|AUTO|INFPersist|INFInite|VARpersist|CLEAR\}}

\text{DISPLAY:PERSistence?}
Related Commands

DISplay VARpersist

Arguments

OFF disables the persistence aspect of the display.

AUTO automatically set the persistence.

INFPersist sets a display mode where any pixels, once touched by samples, remain set until cleared by a mode change.

INFINITE sets a display mode where any pixels, once touched by samples, remain set until cleared by a mode change.

VARPersist sets a display mode where set pixels are gradually dimmed.

CLEAR resets the persist time count down and clears the display of acquired points.

Examples

DISPLAY:PERSISTENCE VARPERSIST sets the persistence aspect of the display to fade set pixels according to the time set in the DISplay:V ARpersist command.

DISPLAY:PERSISTENCE? might return :DISPLAY:PERSISTENCE OFF, indicating that the persistence aspect of the display is disabled.

DISplay:PERSistence:RESET (No Query Form)

This command controls the clearing of persistence data that has been built up over time. Persistence is valid for wave views only.

Group Display Control

Syntax DISplay:PERSistence:RESET

Related Commands DISplay:PERSistence

DISplay:VARpersist

Examples DISplay:PERSistence:RESET clears the display of persistence data.

DISplay:PLoteView<x>:AUTOScale

This command sets or queries the enabled state of autoscale for the specified plot.

Group Display Control
Syntax

\texttt{DISPLAY\textasciitilde PLOTView\textless x\textgreater \textasciitilde AUTOScale \{OFF|ON|\langle NR1\rangle\}}

\texttt{DISPLAY\textasciitilde PLOTView\textless x\textgreater \textasciitilde AUTOScale?}

Arguments

\texttt{PLOTView\textless x\textgreater} is the Plot waveform number.

\texttt{OFF} disables the autoscale feature.

\texttt{ON} enables the autoscale feature.

\texttt{\langle NR1\rangle = 0} disables the autoscale feature; any other value enables the autoscale feature.

Examples

\texttt{DISPLAY\textasciitilde PLOTView1\textasciitilde AUTOScale OFF} turns off plot autoscale.

\texttt{DISPLAY\textasciitilde PLOTView3\textasciitilde AUTOScale?} might return

\texttt{:DISPLAY\textasciitilde PLOTVIEW3\textasciitilde AUTOSCALE 0}, indicating the plot autoscale is off for plot 3.

\texttt{DISPLAY\textasciitilde PLOTView\textless x\textgreater \textasciitilde CURSor\textasciitilde ASOURCE?} (Query Only)

This command queries the cursor source for plot cursor A.

Group

Cursor

Syntax

\texttt{DISPLAY\textasciitilde PLOTView\textless x\textgreater \textasciitilde CURSor\textasciitilde ASOURCE?}

Arguments

\texttt{PLOTView\textless x\textgreater} is the Plot waveform number.

Examples

\texttt{DISPLAY\textasciitilde PLOTView1\textasciitilde CURSor\textasciitilde ASOURCE?} might return

\texttt{:DISPLAY\textasciitilde PLOTVIEW1\textasciitilde CURSOR\textasciitilde ASOURCE PLOT1}, indicating the source of the cursor is plot 1.

\texttt{DISPLAY\textasciitilde PLOTView\textless x\textgreater \textasciitilde CURSor\textasciitilde BSOURCE?} (Query Only)

This command queries the cursor source for plot cursor B.

Group

Cursor

Syntax

\texttt{DISPLAY\textasciitilde PLOTView\textless x\textgreater \textasciitilde CURSor\textasciitilde BSOURCE?}
Arguments  PLOTView<x> is the Plot waveform number.

Examples  DISPLAY:PLOTView1:CURSor:BSOURCe? might return :DISPLAY:PLOTVIEW1:CURSOR:BSOURCE PLOT1 indicating the B cursor source is plot 1.

**DISplay:PLOTView<x>:CURSor:DDT? (Query Only)**

This command returns the delta V over delta T cursor readout value for the specified Plot view.

Group  Cursor

Syntax  DISPLAY:PLOTView<x>:CURSor:DDT?

Arguments  PlotView<x> is the Plot waveform number.


**DISplay:PLOTView<x>:CURSor:FUNCtion**

This command sets or queries the cursor mode for the specified Plot view.

Group  Cursor

Syntax  DISPLAY:PLOTView<x>:CURSor:FUNCtion
{WAVEFORM|VBArS|HBArS|SCREEN}
DISPLAY:PLOTView<x>:CURSor:FUNCtion?

Arguments  PLOTView<x> is the Plot waveform number.

- WAVEFORM specifies to display the paired cursors in YT display format for measuring waveform amplitude and time.
- VBArS specifies vertical bar cursors, which measure in horizontal units.
- HBArS specifies horizontal bar cursors, which measure in vertical units.
SCREEN specifies to display both horizontal and vertical bar cursors, which measure the selected waveform in horizontal and vertical units. Use these cursors to measure anywhere in the waveform display area.

Examples

\texttt{DIsplay:PLOTview1:CURSor:FUNCTION VBARS} displays vertical bar cursors. 
\texttt{DIsplay:PLOTview1:CURSor:FUNCTION?} might return 
\texttt{:DISPLAY:PLOTVIEW1:CURSOR:FUNCTION SCREEN} indicating that both horizontal and vertical bar cursors are displayed.

\textbf{DIsplay:PLOTView<x>:CURSor:HBArs:APOSition}

This command sets or queries the horizontal cursor A position for the specified Plot view.

\begin{itemize}
  \item **Group**: Cursor
  \item **Syntax**: \texttt{DIsplay:PLOTView<x>:CURSor:HBArs:APOSition <NR3>}
  \texttt{DIsplay:PLOTView<x>:CURSor:HBArs:APOSition?}
  \item **Arguments**: 
  \texttt{PLOTView<x>} is the Plot waveform number.
  \texttt{<NR3>} is the cursor position.
  \item **Examples**: 
  \texttt{DIsplay:PLOTView<x>:CURSor:HBArs:APOSition 2.0} sets the cursor to 2 Volts.
  \texttt{DIsplay:PLOTView<x>:CURSor:HBArs:APOSition?} might return 
  \texttt{:DISPLAY:PLOTVIEW1:CURSOR:HBArs:APOSITION 2.2186} indicating the HBArs cursors are at 2.2186 Volts.
\end{itemize}

\textbf{DIsplay:PLOTView<x>:CURSor:HBArs:AUNIts? (Query Only)}

This command queries the horizontal cursor A vertical units for the specified Plot view.

\begin{itemize}
  \item **Group**: Cursor
  \item **Syntax**: \texttt{DIsplay:PLOTView<x>:CURSor:HBArs:AUNIts?}
  \item **Returns**: 
  \texttt{PLOTView<x>} is the Plot waveform number.
\end{itemize}
Examples

**DISPLAY:PLOTView1:CURSor:HBArS:AUnIts?** might return
:DISPLAY:PLOTVIEW1:CURSOR:HBARS:AUNITS "V", indicating the A units are volts.

**DISPLAY:PLOTView<x>:CURSor:HBArS:BPOSition**

This command sets or queries the horizontal cursor B position for the specified Plot view.

**Group**  Cursor

**Syntax**

DISPLAY:PLOTView<x>:CURSor:HBArS:BPOSition <NR3>

DISPLAY:PLOTView<x>:CURSor:HBArS:BPOSition?

**Arguments**

PLOTView<x> is the Plot waveform number.

<NR3> is the HBArS vertical position.

**Examples**

DISPLAY:PLOTView<x>:CURSor:HBArS:BPOSition 2.0 sets the cursor to 2 Volts.

DISPLAY:PLOTView<x>:CURSor:HBArS:BPOSition? might return
:DISPLAY:PLOTVIEW1:CURSOR:HBARS:BPOSITION 2.2186 indicating the HBArS cursors are at 2.2186 Volts.

**DISPLAY:PLOTView<x>:CURSor:HBArS:BUNIts? (Query Only)**

This command queries the cursor B vertical units for the specified Plot view.

**Group**  Cursor

**Syntax**

DISPLAY:PLOTView<x>:CURSor:HBArS:BUNIts??

**Arguments**

PLOTView<x> is the Plot waveform number.

**Examples**

DISPLAY:PLOTView1:CURSor:HBArS:BUNIts? might return
:DISPLAY:PLOTVIEW1:CURSOR:HBARS:BUNITS "V" indicating the units are Volts.
**DISplay:PLOTView<x>:CURSor:HBArs:DELTa? (Query Only)**

This command queries the delta V cursor readout value for the specified Plot view.

**Group**  
Cursor

**Syntax**  
`DISplay:PLOTView<x>:CURSor:HBArs:DELTa??`

**Arguments**  
PLOTView<x> is the Plot waveform number.

**Returns**  
Returns an <NR3> that is the delta V cursor value.

**Examples**  
`DISplay:PLOTView1:CURSor:HBArs:DELTa?` might return  
`:DISPLAY:PLOTVIEW1:CURSOR:HBARS:DELTA 1.2543`, indicating the delta value is 1.2543 V.

**DISplay:PLOTView<x>:CURSor:MODe**

This command sets or queries the cursor tracking mode for the specified Plot view.

**Group**  
Cursor

**Syntax**  
`DISplay:PLOTView<x>:CURSor:MODe {INDEPENDENT|TRACK}`  
`DISplay:PLOTView<x>:CURSor:MODe?`

**Arguments**  
PLOTView<x> is the Plot waveform number.

INDEPENDENT allows independent adjustment of the two cursors.

TRACK ties the navigational functionality of the two cursors together. For cursor A adjustments, this ties the movement of the two cursors together, however, cursor B continues to move independently of cursor A.

**Examples**  
`DISplay:PLOTView1:CURSor:MODe TRACK` sets the cursor to track together.  
`DISplay:PLOTView1:CURSor:MODe?` might return  
`:DISPLAY:PLOTVIEW1:CURSOR:MODE INDEPENDENT`, indicating independent adjustment of the cursors is allowed.
**DISplay:PLOTView<x>:CURSor:ONEOVERDELTATVALUE? (Query Only)**

This command sets or queries the one over delta T cursor readout value for the specified Plot view.

**Group**
Cursor

**Syntax**
DISplay:PLOTView<x>:CURSor:ONEOVERDELTATVALUE?

**Arguments**
PLOTView<x> is the Plot waveform number.

**Returns**
Returns an \(<NR3>\) that is the one over delta T cursor readout value (it may not be delta T, depending on the plot units).

**Examples**
DISplay:PLOTView<x>:CURSor:ONEOVERDELTATVALUE? might return
:DISPLAY:PLOTVIEW1:CURSOR:ONEOVERDELTATVALUE 88.2295E+6
indicating the 1 over delta time value is 88.2295 MHz.

**DISplay:PLOTView<x>:CURSor:SCREEN:AXPOSition**

This command sets or queries the horizontal cursor A position of the specified cursor in the specified view.

**Group**
Cursor

**Syntax**
DISplay:PLOTView<x>:CURSor:SCREEN:AXPOSition <NR3>
DISplay:PLOTView<x>:CURSor:SCREEN:AXPOSition?

**Arguments**
PLOTView<x> is the Plot waveform number.
\(<NR3>\) is the horizontal cursor A position.

**Examples**
DISplay:PLOTView1:CURSor:SCREEN:AXPOSition -5.0 sets the A X cursor position to —5.0 ns.
DISplay:PLOTView1:CURSor:SCREEN:AXPOSition? might return
indicating the cursor A X position is -5.546 ns.
**DISplay:PLOTView<x>:CURSor:SCREEN:AYPOSition**

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

**Group**

Cursor

**Syntax**

```
DISplay:PLOTView<x>:CURSor:SCREEN:AYPOSition <NR3>
DISplay:PLOTView<x>:CURSor:SCREEN:AYPOSition?
```

**Arguments**

PLOTView<x> is the Plot waveform number.

<NR3> is the vertical cursor A position.

**Examples**

```
DISplay:PLOTView1:CURSor:SCREEN:AYPOSition 2.0 set the A cursor Y position to 2.0 Volts.
DISplay:PLOTView1:CURSor:SCREEN:AYPOSition? might return :DISPLAY:PLOTVIEW1:CURSOR:SCREEN:AYPOSITION 1.9035 indicating the A cursor Y position is 1.9035 V.
```

**DISplay:PLOTView<x>:CURSor:SCREEN:BXPOSition**

This command sets or queries the horizontal cursor B position of the specified cursor in the specified view.

**Group**

Cursor

**Syntax**

```
DISplay:PLOTView<x>:CURSor:SCREEN:BXPOSition <NR3>
DISplay:PLOTView<x>:CURSor:SCREEN:BXPOSition?
```

**Arguments**

PLOTView<x> is the Plot waveform number.

<NR3> is the horizontal cursor B position.

**Examples**

```
DISplay:PLOTView1:CURSor:SCREEN:BXPOSition 59.0E-9 sets the B cursor X position to 59.0 ns.
```
DISplay:PLOTView<x>:CURSor:SCREEN:BYPOSition

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group       Cursor

Syntax      DISplay:PLOTView<x>:CURSor:SCREEN:BYPOSition <NR3>
            DISplay:PLOTView<x>:CURSor:SCREEN:BYPOSition?

Arguments   PLOTView<x> is the Plot waveform number.
            <NR3> is the vertical cursor B position.

Examples    DISplay:PLOTView1:CURSor:SCREEN:BYPOSition 0.589 sets the B cursor Y position to 589 mV.
            DISplay:PLOTView1:CURSor:SCREEN:BYPOSition? might return
            indicating the B cursor Y position is 589.0696 mV.

DISplay:PLOTView<x>:CURSor:SPLITMODE

This command sets or queries the cursor source mode in the specified view.

Group       Cursor

Syntax      DISplay:PLOTView<x>:CURSor:SPLITMODE {SAME|SPLIT}
            DISplay:PLOTView<x>:CURSor:SPLITMODE?

Arguments   PLOTView<x> is the Plot waveform number.
            SAME specifies that both cursors are on the same waveform.
            SPLIT specifies that the cursors can be on different waveforms.

Examples    DISplay:PLOTView1:CURSor:SPLITMODE SAME sets the cursors to be on the same waveform.
            DISplay:PLOTView1:CURSor:SPLITMODE? might return
            :DISPLAY:PLOTVIEW1:CURSOR:SPLITMODE SAME indicating both
cursors are on the same waveform.
**Display:PLOTview<x>:CursOr:STATE**

This command sets or queries the visible state of the cursor of the specified cursor in the specified view.

**Group**  
Cursor

**Syntax**  
`DISPLAY:PLOTview<x>:CURSOR:STATE {OFF|ON|<NR1>}`  
`DISPLAY:PLOTview<x>:CURSOR:STATE?`

**Arguments**  
`PLOTview<x>` is the Plot waveform number.

- **OFF**  
  disables the specified cursor.

- **ON**  
  enables the specified cursor.

- `<NR1>`  
  = 0 disables the specified cursor; any other value enables the specified cursor.

**Examples**  
`DISPLAY:PLOTview1:CURSOR:STATE OFF`  
 disables the specified cursor.

`DISPLAY:PLOTview5:CURSOR:STATE?`  
 might return

`:DISPLAY:PLOTVIEW5:CURSOR:STATE 1` indicating cursors are on.

**Display:PLOTview<x>:CURSOr:VBarS:APOSiOn**

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

**Group**  
Cursor

**Syntax**  
`DISPLAY:PLOTview<x>:CURSOR:VBARs:APOSITION <NR3>`  
`DISPLAY:PLOTview<x>:CURSOR:VBARs:APOSITION?`

**Arguments**  
`PLOTview<x>` is the Plot waveform number.

- `<NR3>`  
  is the vertical cursor A position.

**Examples**  
`DISPLAY:PLOTview1:CURSOR:VBARs:APOSITION 50.0E-9`  
 sets the VBARs APOSITION to 50 ns.
Disp:PLotView1:CURSor:VBArS:APOSIOn? might return
:Disp:PLotView1:CURSor:VBArS:APOSIOn -60.0796E-9
indicating the VBArS APOSIOn is set to -60.08 ns.

Disp:PLotView<x>:CURSor:VBArS:BPOSIOn

This command sets or queries the vertical cursor B position of the specified cursor
in the specified view.

Group       Cursor

Syntax       Disp:PLotView<x>:CURSor:VBArS:BPOSIOn <NR3>
             Disp:PLotView<x>:CURSor:VBArS:BPOSIOn?

Arguments    PLOTvew<x> is the Plot waveform number.
             <NR3> is the vertical cursor B position.

Examples     :Disp:PLotView1:CURSor:VBArS:BPOSIOn 50.0E-9 sets the VBArS
             BPOSIOn to 50 ns.
             :Disp:PLotView1:CURSor:VBArS:BPOSIOn? might return
             :Disp:PLotView1:CURSor:VBArS:BPOSIOn -60.0796E-9
             indicating the VBArS BPOSIOn is set to 60.08 ns.

Disp:PLotView<x>:CURSor:VBArS:DELTa? (Query Only)

This command queries the delta T cursor readout value of the specified cursor
in the specified view.

Group       Cursor

Syntax       Disp:PLotView<x>:CURSor:VBArS:DELTa?

Arguments    PLOTvew<x> is the Plot waveform number.

Returns      The delta T cursor readout value is returned.
Examples

DisPlay:PLOTView1:CURSoR:VBArS:DELTa? might return
:DISPLAY:PLOTVIEW1:CURSOR:VBARS:DELTA 119.7593E-9 indicating
the VBARS DELTA is set to 119.76 ns.

DisPlay:PLOTView<x>:CURSoR:VBArS:UNIts? (Query Only)

This command queries the VBArS cursor readout units of the specified cursor
in the specified view.

Group  Cursor

Syntax  DisPlay:PLOTView<x>:CURSoR:VBArS:UNIts?

Arguments  PLOTView<x> is the Plot waveform number.

Returns  The VBArS cursor readout units are returned.

Examples  DisPlay:PLOTView<x>:CURSoR:VBArS:UNIts? might return
:DISPLAY:PLOTVIEW5:CURSOR:VBARS:UNITS "V" indicating the units are
Volts.

DisPlay:PLOTView<x>:CURSoR:WAVEform:APOSition

This command sets or queries the waveform cursor A horizontal position of the
specified cursor in the specified view.

NOTE. In case of XY plot, this command has no effect when used to set the value.
In case of bathtub plot, this command sets or returns the cursor A vertical position.
For all other plots, this command sets or returns the cursor A horizontal position.

Group  Cursor

Syntax  DisPlay:PLOTView<x>:CURSoR:WAVEform:APOSition <NR3>
DisPlay:PLOTView<x>:CURSoR:WAVEform:APOSition?

Arguments  PLOTView<x> is the Plot waveform number.
<NR3> is the horizontal cursor A position.
Examples  

**Display:**PLOTView1:CURSor:WAVEform:APOSition -50.0E-9 sets the waveform APOSITION to -50.0 ns.

**Display:**PLOTView1:CURSor:WAVEform:APOSition? might return :DISPLAY:PLOTVIEW1:CURSOR:WAVEFORM:APOSITION -60.0796E-9 indicating the waveform APOSITION is set to -60.08 ns.

**Display:**PLOTView<x>:CURSor:WAVEform:BPOSition

This command sets or queries the waveform cursor B horizontal position of the specified cursor in the specified view.

**NOTE.** In case of XY plot, this command has no effect when used to set the value. In case of bathtub plot, this command has no effect. Query returns invalid values. For all other plots, this command sets or returns the cursor B horizontal position.

**Group**  
Cursor

**Syntax**  
**Display:**PLOTView<x>:CURSor:WAVEform:BPOSition <NR3>  
**Display:**PLOTView<x>:CURSor:WAVEform:BPOSition?

**Arguments**  
PLOTView<x> is the Plot waveform number.  
<NR3> is the horizontal cursor B position.

**Examples**  
**Display:**PLOTView1:CURSor:WAVEform:BPOSition 3.0 sets the position to 3.0 V.

**Display:**PLOTView5:CURSor:WAVEform:BPOSition? might return :DISPLAY:PLOTVIEW5:CURSOR:WAVEFORM:BPOSITION 2.600517840 indicating the position is 2.5 V.

**Display:**PLOTView<x>:GRIDlines

This command sets or queries the Grid (graticule) lines setting of the specified plot. This command works for plots that have vertical and horizontal units associated with the graticule. For example, this command does not work for XY or XYZ plots.

**Group**  
Display Control
Syntax
DISPLAY:PLOTView<x>:GRIDlines {HORizontal|VERTical|BOTH}
DISPLAY:PLOTView<x>:GRIDlines?

Arguments
PLOTView<x> is the Plot waveform number.
HORizontal specifies horizontal grid lines.
VERTical specifies vertical grid lines.
BOTH specifies both vertical and horizontal grid lines.

Examples
DISPLAY:PLOTView1:GRIDlines Vertical sets the vertical grid lines to display.
DISPLAY:PLOTView1:GRIDlines? might return :DISPLAY:PLOTVIEW1:GRIDLINES BOTH indicating that both vertical and horizontal grid lines are displayed.

DISPLAY:PLOTVIEW<x>:XAXIS:SCALE
This command sets or queries the horizontal scale setting for applicable plots (LINEAR or LOG) for the specified plot view.

Group
Display

Syntax
DISPLAY:PLOTVIEW<x>:XAXIS:SCALE {LINEAR|LOG}

Arguments
PLOTView<x> is the Plot waveform number.
LINEAR creates a plot with linear scales.
LOG creates a plot with logarithmic scales.

Examples
DISPLAY:PLOTVIEW1:XAXIS:SCALE LOG creates a plot with a logarithmic horizontal scale.

DISPLAY:PLOTVIEW<x>:YAXIS:SCALE
This command sets or queries the vertical scale setting for applicable plots (LINEAR or LOG) in the specified plot view.
Group: Display

Syntax: DISPLAY:PLOTVIEW<x>:YAXIS:SCALE {LINEAR|LOG}

Arguments: PLOTview<x> is the Plot waveform number.
LINEAR specifies a linear vertical scale.
LOG specifies a logarithmic vertical scale.

Examples:
DISPLAY:PLOTVIEW1:YAXIS:SCALE LINEAR sets the plot to use a linear vertical scale.

Group: Zoom

Syntax: DISPLAY:PLOTVIEW<x>:ZOOM:XAXIS:FROM <NR3>
DISPLAY:PLOTVIEW<x>:ZOOM:XAXIS:FROM?

Arguments: PLOTview<x> is the Plot waveform number.
<NR3> is start of the zoom x-axis.

Examples:
DISPLAY:PLOTVIEW1:ZOOM:XAXIS:FROM 2.0 sets the start of the zoom x-axis to 2.0 V.

Group: Zoom

Syntax: DISPLAY:PLOTVIEW<x>:ZOOM:XAXIS:TO

Arguments: PLOTview<x> is the Plot waveform number.

Examples:
DISPLAY:PLOTVIEW1:ZOOM:XAXIS:TO 2.0 sets the start of the zoom x-axis to 2.0 V.
DISPLAY:PLOTVIEW3:ZOOM:XAXIS:FROM 1.9844803459459 indicating the start of the zoom x-axis is 1.9845 V.
Group: Zoom

Syntax:
- **DISplay:PLOTView<x>:ZOOM:XAXIS:TO** <NR3>
- **DISplay:PLOTView<x>:ZOOM:XAXIS:TO?**

Arguments:
- **PLOTView<x>** is the Plot waveform number.
- **<NR3>** is the end of the zoom x-axis.

Examples:
- **DISplay:PLOTView1:ZOOM:XAXIS:TO 2.5** sets the end of the zoom x-axis to 2.5 V.
- **DISplay:PLOTView3:ZOOM:XAXIS:TO?** might return **DISPLAY:PLOTVIEW3:ZOOM:XAXIS:TO 2.2144883507508** indicating the end of the zoom x-axis is 2.21 V.

**DISplay:PLOTView<x>:ZOOM:YAXIS:FROM**

This command sets or queries the bottom value of the zoom y-axis in the specified plot view.

Group: Zoom

Syntax:
- **DISplay:PLOTView<x>:ZOOM:YAXIS:FROM** <NR3>
- **DISplay:PLOTView<x>:ZOOM:YAXIS:FROM?**

Arguments:
- **PLOTView<x>** is the Plot waveform number.
- **<NR3>** is the bottom value of the zoom y-axis.

Examples:
- **DISplay:PLOTView1:ZOOM:YAXIS:FROM -1.0E+20** sets the bottom value of the zoom y-axis to -1E+20.
- **DISplay:PLOTView1:ZOOM:YAXIS:FROM?** might return **DISPLAY:PLOTVIEW1:ZOOM:YAXIS:FROM -1.0E+21** indicating the bottom value of the zoom y-axis is -1E+21.

**DISplay:PLOTView<x>:ZOOM:YAXIS:TO**

This command sets or queries the top value of the zoom y-axis in the specified plot view.
Group  Zoom

Syntax  
```
DISPLAY:PLOTVIEW<x>:ZOOM:YAXIS:TO <NR3>
DISPLAY:PLOTVIEW<x>:ZOOM:YAXIS:TO?
```

Arguments  
- `PLOTVIEW<x>` is the Plot waveform number.
- `<NR3>` is the top value of the zoom y-axis.

Examples  
- `DISPLAY:PLOTVIEW1:ZOOM:YAXIS:TO 100` sets the top value of the zoom y-axis to 100.
- `DISPLAY:PLOTVIEW1:ZOOM:YAXIS:TO?` might return 
  `:DISPLAY:PLOTVIEW3:ZOOM:YAXIS:FROM 0.0E+0` indicating the top value of the zoom y-axis is 0.0 hits (for a histogram plot, or it could be Volts for an XY plot).

**DISPLAY:REFFFTView<x>:AUTOScale**

This command sets or queries the enabled state of auto-scale for plots.

Group  Display Control

Syntax  
```
DISPLAY:REFFFTView<x>:AUTOScale {OFF|ON|0|1}
DISPLAY:REFFFTView<x>:AUTOScale?
```

Arguments  
- `REFFFTView<x>` is the plot number.
- `<NR1>` = 0 disables auto-scale on the specified reffftview, any other value turns this feature on.
  - OFF disables auto-scale on the specified reffftview.
  - ON enables the specified channel on the specified waveview.

Examples  
- `DISPLAY:REFFFTView1:AUTOScale 1` enables auto-scale on the specified view.
- `DISPLAY:REFFFTView5:AUTOScale?` might return 
  `:DISPLAY:REFFFTVIEW5:AUTOSCALE 1` indicating auto-scale is on.
**DISplay:REFFFTView<x>:CURSor:ASOUrce? (Query Only)**

This command returns the cursor source for plot cursor A.

**Group** Cursor

**Syntax** 
\[ \text{DISplay}:\text{REFFFTView}<x>:\text{CURSor}:\text{ASOUrce}? \]

**Arguments** 
\[ \text{PLOTView}<x> \] is the Plot waveform number.

**Returns** Returns the cursor source for plot cursor A.

**Examples** 
\[ \text{DISplay}:\text{REFFFTView}5:\text{CURSor}:\text{ASOUrce}? \] might return 
\[ \text{DISplay}:\text{REFFFTView}5:\text{CURSor}:\text{ASOUrce} \text{ REF5} \] indicating the A cursor source is reference 5.

**DISplay:REFFFTView<x>:CURSor:BSOUrce? (Query Only)**

This command returns the cursor source for plot cursor B.

**Group** Cursor

**Syntax** 
\[ \text{DISplay}:\text{REFFFTView}<x>:\text{CURSor}:\text{BSOUrce}? \]

**Returns** Returns the cursor source for plot cursor B.

**Examples** 
\[ \text{DISplay}:\text{REFFFTView}5:\text{CURSor}:\text{BSOUrce}? \] might return 
\[ \text{DISplay}:\text{REFFFTView}5:\text{CURSor}:\text{BSOUrce} \text{ REF5} \] indicating the source of the B cursor is reference 5.

**DISplay:REFFFTView<x>:CURSor:DDT? (Query Only)**

This command returns the delta V over delta T cursor readout value of the specified cursor in the specified view.

**Group** Cursor
Syntax: DISplay:REFFFTView<x>:CURSor:DDT?

Returns: Returns the delta V over delta T cursor readout value.

Examples: DISplay:REFFFTView5:CURSor:DDT? might return
:DISPLAY:REFFFTVIEW5:CURSOR:DDT  9.91E+37 indicating the delta V over
delta T cursor readout value is 9.91E+37.

**DISplay:REFFFTView<x>:CURSor:FUNCTION**

This command sets or queries the cursor type of the specified cursor in the
specified view.

Group: Cursor

Syntax: DISplay:REFFFTView<x>:CURSor:FUNCTION
{WAVEform|VBArs|HBArs|SCREEN}
DISplay:REFFFTView<x>:CURSor:FUNCTION?

Arguments:
- HBArs specifies horizontal bar cursors, which measure in vertical units.
- VBArs specifies vertical bar cursors, which measure in horizontal units.
- SCREEN specifies both horizontal and vertical bar cursors, which measure in
  horizontal and vertical units specified by the cursor sources. Use these cursors
to measure anywhere in the waveform display area.
- WAVEform specifies paired or split cursors in YT display format for measuring
  waveform amplitude and time. In XY and XYZ format, these cursors indicate the
  amplitude positions of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis
  and Ch2 is the Y axis) relative to the trigger.

Examples: DISplay:REFFFTView5:CURSor:FUNCTION screen sets the cursor function
to screen.
DISplay:REFFFTView5:CURSor:FUNCTION? might return
:DISPLAY:REFFFTVIEW5:CURSOR:FUNCTION WAVEFORM indicating the cursor
function is set to waveform.

**DISplay:REFFFTView<x>:CURSor:HBArs:APOSition**

This command sets or queries the vertical cursor A position of the specified cursor
in the specified view.
Group: Cursor

Syntax:

```
DISPLAY:REFFFTView<x>:CURSor:HBArs:APOSition <NR3>
DISPLAY:REFFFTView<x>:CURSor:HBArs:APOSition?
```

Arguments:

- `<NR3>` is the vertical cursor A position of the specified cursor in the specified view.

Examples:

- `DISPLAY:REFFFTView5:CURSor:HBArs:APOSition -20` sets the A cursor position to -20.

**DISPLAY:REFFFTView<x>:CURSor:HBArs:AUNIts? (Query Only)**

This command returns cursor A vertical units of the specified cursor in the specified view.

Group: Cursor

Syntax:

```
DISPLAY:REFFFTView<x>:CURSor:HBArs:AUNIts?
```

Returns:

Returns the cursor A vertical units of the specified cursor in the specified view.

Examples:

- `DISPLAY:REFFFTView5:CURSor:HBArs:AUNIts?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:HBARS:AUNITS "dBm"` indicating the A cursor units are dBm.

**DISPLAY:REFFFTView<x>:CURSor:HBArs:BPOSition**

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group: Cursor

Syntax:

```
DISPLAY:REFFFTView<x>:CURSor:HBArs:BPOSition <NR3>
DISPLAY:REFFFTView<x>:CURSor:HBArs:BPOSition?
```
Arguments  

<NR3> is the vertical cursor B position of the specified cursor in the specified view.

Examples  

DISPLAY:REFF7View5:CURSor:HBArs:BPOSition -90 sets the B cursor position to -90.


**DISPLAY:REFF7View<x>:CURSor:HBArs:BUNIts? (Query Only)**

This command returns the cursor B vertical units of the specified cursor in the specified view.

Group  

Cursor

Syntax  

DISPLAY:REFF7View<x>:CURSor:HBArs:BUNIts?

Returns  

Returns the cursor B vertical units of the specified cursor in the specified view.

Examples  

DISPLAY:REFF7View5:CURSor:HBArs:BUNIts? might return :DISPLAY:REFF7VIEW5:CURSOR:HBARS:BUNITS "dBm" indicating the units are dBm.

**DISPLAY:REFF7View<x>:CURSor:HBArs:DELTa? (Query Only)**

This command returns the delta V cursor readout value of the specified cursor in the specified view.

Group  

Cursor

Syntax  

DISPLAY:REFF7View<x>:CURSor:HBArs:DELTa?

Returns  

Returns the delta V cursor readout value of the specified cursor in the specified view.

Examples  

**DISplay:REFFFTView<x>:CURSor:MODE**

This command sets or queries the cursor tracking mode of the specified cursor in the specified view.

**Group**  
Cursor

**Syntax**  
```
DISplay:REFFFTView<x>:CURSor:MODE {INDEPENDENT|TRACK}
DISplay:REFFFTView<x>:CURSor:MODE?
```

**Arguments**  
**TRACK** ties the navigational functionality of the two cursors together. For cursor A adjustments, this ties the movement of the two cursors together, however, cursor B continues to move independently of cursor A.

**Examples**  
```
DISplay:REFFFTView5:CURSor:MODE TRACK sets the cursors to track together.

DISplay:REFFFTView5:CURSor:MODE? might return
:DISPLAY:REFFFTVIEW5:CURSOR:MODE INDEPENDENT indicating the cursors move independently.
```

**DISplay:REFFFTView<x>:CURSor:ONEOVERDELTATVALUE? (Query Only)**

This command returns the one over delta T cursor readout value of the specified cursor in the specified view.

**Group**  
Cursor

**Syntax**  
```
DISplay:REFFFTView<x>:CURSor:ONEOVERDELTATVALUE?
```

**Examples**  
```
DISplay:REFFFTView5:CURSor:ONEOVERDELTATVALUE? might return
:DISPLAY:REFFFTVIEW5:CURSOR:ONEOVERDELTATVALUE 2.540935140340E-9 indicating the one over delta T cursor readout value is 2.54 ns.
```

**DISplay:REFFFTView<x>:CURSor:SCREEN:AXPOSition**

This command sets or queries the horizontal cursor A position of the specified cursor in the specified view.
Commands listed in alphabetical order

**Group**  
**Cursor**

**Syntax**  
DISplay:REFFFTView<x>:CURSor:SCREEN:AXPOSition <NR3>  
DISplay:REFFFTView<x>:CURSor:SCREEN:AXPOSition?

**Arguments**  
<NR3> is the horizontal cursor A position of the specified cursor in the specified view.

**Examples**  
DISplay:REFFFTView5:CURSor:SCREEN:AXPOSition 1.0E+6 sets the cursor position to 1.0 MHz.  
DISplay:REFFFTView5:CURSor:SCREEN:AXPOSition? might return :DISPLAY:REFFFTVIEW5:CURSOR:SCREEN:AXPOSITION 792.4465962305570E+3 indicating the A cursor horizontal position is 792.447 kHz.

**DISplay:REFFFTView<x>:CURSor:SCREEN:AYPOSition**

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

**Group**  
**Cursor**

**Syntax**  
DISplay:REFFFTView<x>:CURSor:SCREEN:AYPOSition <NR3>  
DISplay:REFFFTView<x>:CURSor:SCREEN:AYPOSition?

**Arguments**  
<NR3> is the vertical cursor A position of the specified cursor in the specified view.

**Examples**  
DISplay:REFFFTView5:CURSor:SCREEN:AYPOSition -20.9 sets the position to -20 dBm.  

**DISplay:REFFFTView<x>:CURSor:SCREEN:BXPOSition**

This command sets or queries the horizontal cursor B position of the specified cursor in the specified view.
Group  Cursor

Syntax  DISPLAY:REFFFTView<x>:CURSor:SCREEN:BXPOsition <NR3>
        DISPLAY:REFFFTView<x>:CURSor:SCREEN:BXPOsition?

Arguments  <NR3> is the horizontal cursor B position of the specified cursor in the specified view.

Examples  DISPLAY:REFFFTView5:CURSor:SCREEN:BXPOsition 300 sets the position to 300 MHz.

**DISPLAY:REFFFTView<x>:CURSor:SCREEN:BYPOsition**

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group  Cursor

Syntax  DISPLAY:REFFFTView<x>:CURSor:SCREEN:BYPOsition <NR3>
        DISPLAY:REFFFTView<x>:CURSor:SCREEN:BYPOsition?

Arguments  <NR3> is the vertical cursor B position of the specified cursor in the specified view.

Examples  DISPLAY:REFFFTView5:CURSor:SCREEN:BYPOsition -80 sets the cursor position to -90 dBm.

**DISPLAY:REFFFTView<x>:CURSor:SPLITMODE**

This command sets or queries whether both cursors have same or different source.

Group  Cursor
**Syntax**

```
DISPLAY:REFFFTView<x>:CURSor:SPLITMODE {SAME|SPLIT}
DISPLAY:REFFFTView<x>:CURSor:SPLITMODE?
```

**Arguments**

SAME specifies both cursors have the same sources.

SPLIT specifies both cursors have different sources.

**Examples**

```
DISPLAY:REFFFTView5:CURSor:SPLITMODE SPLIT specified that the cursors have different sources.

DISPLAY:REFFFTView5:CURSor:SPLITMODE? might return :DISPLAY:REFFFTVIEW5:CURSOR:SPLITMODE SAME indicating both cursors have the same source.
```

**DISPLAY:REFFFTView<x>:CURSor:STATE**

This command sets or queries the visible state of the cursor of the specified cursor in the specified view.

**Group**

Cursor

**Syntax**

```
DISPLAY:REFFFTView<x>:CURSor:STATE {OFF|ON|0|1}
DISPLAY:REFFFTView<x>:CURSor:STATE?
```

**Arguments**

<NR1> = 0 specifies the cursor is not visible; any other value displays the cursor.

OFF specifies the cursor is not visible.

ON displays the cursor.

**Examples**

```
DISPLAY:REFFFTView5:CURSor:STATE ON specifies the cursor is visible.

DISPLAY:REFFFTView5:CURSor:STATE? might return :DISPLAY:REFFFTVIEW5:CURSOR:STATE 1 indicating the cursor is visible.
```

**DISPLAY:REFFFTView<x>:CURSor:VBArs:APOSition**

This command sets or queries the horizontal cursor A position of the specified cursor in the specified view.

**Group**

Cursor
Syntax

```
DISPLAY:REFFFTView<x>:CURSor:VBArs:APOSITION <NR3>
DISPLAY:REFFFTView<x>:CURSor:VBArs:APOSITION?
```

Arguments

<NR3> is the horizontal cursor A position of the specified cursor in the specified view.

Examples

```
DISPLAY:REFFFTView5:CURSor:VBArs:APOSITION 1.0E+6 sets the cursor to 1 MHz.
DISPLAY:REFFFTView5:CURSor:VBArs:APOSITION? might return :DISPLAY:REFFFTVIEW5:CURSOR:VBARS:APOSITION 792.4465962305570E+3 indicating the cursor position is 792.447 kHz.
```

**DISplay:REFFFTView<x>:CURSor:VBArs:BPOSition**

This command sets or queries the horizontal cursor B position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISPLAY:REFFFTView<x>:CURSor:VBArs:BPOSITION <NR3>
DISPLAY:REFFFTView<x>:CURSor:VBArs:BPOSITION?
```

Arguments

<NR3> is the horizontal cursor B position of the specified cursor in the specified view.

Examples

```
DISPLAY:REFFFTView5:CURSor:VBArs:BPosition 300.0E+6 sets the cursor position to 300 MHz.
```

**DISplay:REFFFTView<x>:CURSor:VBArs:DELTa? (Query Only)**

This command returns the delta T cursor readout value of the specified cursor in the specified view.

Group

Cursor
Syntax  DISplay:REFFFTView<x>:CURSor:VBArs:DELTa?

Returns  Returns the delta T cursor readout value of the specified cursor in the specified view.

Examples  DISplay:REFFFTView5:CURSor:VBArs:DELTa? might return
          :DISPLAY:REFFFTVIEW5:CURSOR:VBARS:DELTA 393.558937038906E+6
          indicating the delta T cursor readout value is 393.556 MHz.

DISplay:REFFFTView<x>:CURSor:VBArs:UNIts? (Query Only)

This command returns cursor A vertical units of the specified cursor in the specified view.

Group  Cursor

Syntax  DISplay:REFFFTView<x>:CURSor:VBArs:UNIts?

Returns  Returns cursor A vertical units of the specified cursor in the specified view.

Examples  DISplay:REFFFTView5:CURSor:VBArs:UNIts? might return
          :DISPLAY:REFFFTVIEW5:CURSOR:VBARS:UNITS "Hz" indicating the A
          cursor vertical units are Hz.

DISplay:REFFFTView<x>:CURSor:WAVEform:AHPOSition? (Query Only)

This query-only command returns the value of the cursor A horizontal position.

Group  Cursor

Syntax  DISplay:REFFFTView<x>:CURSor:WAVEform:AHPOSition?

Examples  DISplay:REFFFTView1:CURSor:WAVEform:AHPOSition? might return
**DISplay:REFFFTView<x>:CURSor:WAVEform:APOSition**

Sets or returns the waveform cursor A position in the specified plot view.

*NOTE. In case of an XY plot, this command has no effect when used to set the value. In the case of a bathtub plot, this command sets or queries the cursor A vertical position. For all other plots, this command sets or queries the cursor A horizontal position.*

**Group**  
Cursor

**Syntax**

```
DISplay:REFFFTView<x>:CURSor:WAVEform:APOSition <NR3>
DISplay:REFFFTView<x>:CURSor:WAVEform:APOSition?
```

**Arguments**

<NR3> is the waveform cursor A position in the specified plot view.

**Examples**

```
DISplay:REFFFTView5:CURSor:WAVEform:APOSition 7E+5 sets the position to 700 kHz.

DISplay:REFFFTView5:CURSor:WAVEform:APOSition? might return :DISPLAY:REFFFTVIEW5:CURSOR:WAVEFORM:APOSITION 792.4465962305570E+3 indicating the A cursor waveform position is 792.4 kHz.
```

**DISplay:REFFFTView<x>:CURSor:WAVEform:AVPOSition? (Query Only)**

This query-only command returns the value of the cursor A vertical position.

**Group**  
Cursor

**Syntax**

```
DISplay:REFFFTView<x>:CURSor:WAVEform:AVPOSition?
```

**Examples**

```
DISplay:REFFFTView<x>:CURSor:WAVEform:AVPOSition? might return :DISPLAY:REFFFTVIEW1:CURSOR:WAVEFORM:AVPOSition 1.605E+0 indicating the cursor vertical position is 1.6 dBm.
```

**DISplay:REFFFTView<x>:CURSor:WAVEform:BHPOSition? (Query Only)**

This query-only command returns the value of the cursor B horizontal position.
Group   Cursor

Syntax   DISPLAY:REFFFTView<x>:CURSor:WAVEform:BHPOSition?

Examples  DISPLAY:REFFFTView<x>:CURSor:WAVEform:BHPOSition? might return

**DISPLAY:REFFFTView<x>:CURSor:WAVEform:BPOSition**

Sets or returns the waveform cursor B position in the specified plot view.

**NOTE.** In case of an XY plot, this command has no effect when used to set the value. In the case of a bathtub plot, this command has no effect. Queries return invalid values. For all other plots, this command sets or queries the cursor B horizontal position.

Group   Cursor

Syntax   DISPLAY:REFFFTView<x>:CURSor:WAVEform:BPOSition <NR3>
          DISPLAY:REFFFTView<x>:CURSor:WAVEform:BPOSition?

Arguments  <NR3> is the waveform cursor B position in the specified plot view

Examples  DISPLAY:REFFFTView5:CURSor:WAVEform:BPOSition 4E+8 sets the cursor position to 400 MHz.
           DISPLAY:REFFFTView5:CURSor:WAVEform:BPOSition? might return
           :DISPLAY:REFFFTVIEW5:CURSOR:WAVEFORM:BPOSITION 394.3483403001212E+6 indicating the B waveform cursor position is 394.35 MHz.

**DISPLAY:REFFFTView<x>:CURSor:WAVEform:BVPOSition? (Query Only)**

This query-only command returns the value of the cursor B vertical position.

Group   Cursor

Syntax   DISPLAY:REFFFTView<x>:CURSor:WAVEform:BVPOSition?
Examples  

```
:DISPlay:REFFFTView1:CURSor:WAVEform:BVPOsition? might
return :DISPlay:REFFFTView<x>:CURSor:WAVEform:BVPOsition
-119.866E+0 indicating the cursor vertical position is -119.9 dBm.
```

### DISplay:REFFFTView<x>:GRIDlines

This command sets or returns the grid lines setting of the plot.

**Group**  
Display Control

**Syntax**  
```
DISPlay:REFFFTView<x>:GRIDlines {HORizontal|VERTical|BOTH}
DISPlay:REFFFTView<x>:GRIDlines?
```

**Arguments**  
- **HORizontal** specifies horizontal grid lines.
- **VERTical** specifies vertical grid lines.
- **BOTH** specifies both horizontal and vertical grid lines.

**Examples**  
```
:DISPlay:REFFFTView5:GRIDlines hhorizontal specifies horizontal gird
lines.
:DISPlay:REFFFTView5:GRIDlines?  might return
:DISPLAY:REFFFTVIEW5:GRIDLINES BOTH indicating that both vertical
and horizontal grid lines are displayed
```

### DISplay:REFFFTView<x>:REF:REF<x>:STATE

This command sets or queries the state of the specified reference waveform in
the specified waveview.

**Group**  
Display Control

**Syntax**  
```
DISPlay:REFFFTView<x>:REF:REF<x>:STATE boolean
DISPlay:REFFFTView<x>:REF:REF<x>:STATE?
```

**Arguments**  
- `<NR1>` = 0 disables the specified reference; any other value turns this feature on.
- **OFF** disables the display the specified reference.
- **ON** enables the specified reference.
Examples

DISplay:REFFFTView5:REF:REF5:STATE 0 turns display of the reference off.

**DISplay:REFFFTView<x>:XAXIS:SCALE**

This command sets or queries the x-axis scale setting for Ref FFT.

**Group**
Display Control

**Syntax**
DISplay:REFFFTView<x>:XAXIS:SCALE {LINEAr|LOG}
DISplay:REFFFTView<x>:XAXIS:SCALE?

**Arguments**
- **LINEAr** specifies a linear scale.
- **LOG** specifies a logarithmic scale.

**Examples**

DISplay:REFFFTView5:XAXIS:SCALE linear sets the x-axis scale to linear.

**DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM**

This command sets or returns the left edge of the zoom x-axis in the specified plot view.

**Group**
Zoom

**Syntax**
DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM <NR3>
DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM?

**Arguments**
- **<NR3>** is the left edge of the zoom x-axis in the specified plot view.

**Examples**

DISplay:REFFFTView5:ZOOM:XAXIS:FROM 2.0E+6 sets the left edge of the specified view to 2.0 MHz.
DISplay:REFFFTView5:ZOOM:XAXIS:FROM? might return
:DISPLAY:REFFFTVIEW5:ZOOM:XAXIS:FROM 1.3295739742362E+6 indicating the left edge of the specified view is 1.33 MHz.

**DISplay:REFFFTView<x>:ZOOM:XAXIS:TO**

This command sets or queries the right edge of the zoom x-axis in the specified plot view.

**Group**  
Zoom

**Syntax**  
DISplay:REFFFTView<x>:ZOOM:XAXIS:TO <NR3>  
DISplay:REFFFTView<x>:ZOOM:XAXIS:TO?

**Arguments**  
<NR3> is the right edge of the zoom x-axis in the specified plot view.

**Examples**  
DISplay:REFFFTView5:ZOOM:XAXIS:TO 2.0E+8 sets the right edge of the zoom x-axis to 200 MHz.  
DISplay:REFFFTView5:ZOOM:XAXIS:TO? might return  
:DISPLAY:REFFFTVIEW5:ZOOM:XAXIS:TO 235.0376933178995E+6 indicating the right edge of the zoom x-axis is 235.0 MHz.

**DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM**

This command sets or queries the bottom value of the zoom y-axis in the specified plot view.

**Group**  
Zoom

**Syntax**  
DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM <NR3>  
DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM?

**Arguments**  
<NR3> is the bottom value of the zoom y-axis in the specified plot view.

**Examples**  
DISplay:REFFFTView5:ZOOM:YAXIS:FROM -120.0 sets the bottom value of the zoom y-axis to -120.0 dBm.
Display:REFFTView<x>:ZOOM:YAXIS:FROM

This command sets or queries the top value of the zoom y-axis in the specified plot view.

**Group**  
Zoom

**Syntax**  
`DISPLAY:REFFTView<x>:ZOOM:YAXIS:FROM`  
`DISPLAY:REFFTView<x>:ZOOM:YAXIS:FROM <NR3>`

**Arguments**  
`<NR3>` is the top value of the zoom y-axis in the specified plot view.

**Examples**  
`DISPLAY:REFFTView5:ZOOM:YAXIS:TO 0.0` sets the top value of the zoom y-axis to 0.0 dBm.

`DISPLAY:REFFTView5:ZOOM:YAXIS:TO?` might return

`:DISPLAY:REFFTView5:ZOOM:YAXIS:TO -129.7533120632172`
indicating the bottom value of the zoom y-axis is -129.8 dBm.

Display:REFFTView<x>:ZOOM:YAXIS:TO

This command sets or queries the top value of the zoom y-axis in the specified plot view.

**Group**  
Zoom

**Syntax**  
`DISPLAY:REFFTView<x>:ZOOM:YAXIS:TO <NR3>`

**Arguments**  
`<NR3>` is the top value of the zoom y-axis in the specified plot view.

**Examples**  
`DISPLAY:REFFTView5:ZOOM:YAXIS:TO 0.0` sets the top value of the zoom y-axis to 0.0 dBm.

`DISPLAY:REFFTView5:ZOOM:YAXIS:TO?` might return

`:DISPLAY:REFFTView5:ZOOM:YAXIS:TO 15.8820685863495`
indicating the top value of the zoom y-axis is 15.9 dBm.

Display:SELEc:t:BUS

This command sets or queries the overall selected bus. Sets are applied to all views that contain the source and the selected view is changed. When multiple buses are open, querying the command gives the correct result, but the bus cannot set.

**Group**  
Display Control

**Syntax**  
`DISPLAY:SELEc:t:BUS BUS<x>`

**Arguments**  
`BUS<x>` is the selected bus.

**Examples**  
`DISPLAY:SELEc:t:BUS Bus2` sets Bus 2 as the selected bus.

`DISPLAY:SELEc:t:BUS?` might return

`:DISPLAY:SELEc:t:BUS BUS1`
indicating Bus 1 is the selected bus.
DISplay:SELect:MATH

This command sets or queries the overall selected math. Sets are applied to all views that contain the source and the selected view is changed. When multiple Math are open, querying the command gives the correct result, but the required Math cannot be set.

Group Display Control

Syntax DISplay:SELect:MATH MATH<x>
      DISplay:SELect:MATH?

Arguments MATH<x> is the selected math.

Examples DISplay:SELect:MATH MATH1 selects Math 1.
         DISplay:SELect:MATH? might display :DISPLAY:SELECT:MATH MATH2 indicating that Math 2 is selected.

DISplay:SELect:REFerence

This command sets or queries the overall selected reference waveform. Sets are applied to all views that contain the source and the selected view is changed.

Group Display Control

Syntax DISplay:SELect:REFerence {NONE|REF<x>}

Arguments Arguments are the selected reference.


DISplay:SELect:SOUrce

This command sets or queries the overall selected source. Sets are applied to all views that contain the source and the selected view is changed.
Group          Display Control

Syntax   DISPLAY:SELECT:SOURce
          {NONE|CH<x>|BUS<x>|MATH<x>|PLOT<x>|REF<x>}

Arguments   Arguments are the selected source.

Examples   DISPLAY:SELECT:SOURce Bus1 selects Bus1 as the selected source.
            DISPLAY:SELECT:SOURce? might return DISPLAY:SELECT:SOURce MATH1 indicating Math 1 is the selected source.

DISPLAY:SELECT:VIEW

This command sets or queries the selected view.

Group          Display Control

Syntax   DISPLAY:SELECT:VIEW
          {WAVEVIEW1|MATHFFT<x>|PLOTVIEW<x>|REFFFT<x>}

Arguments   Arguments are the selected view.

Examples   DISPLAY:SELECT:VIEW PLOTVIEW1 selects PLOTVIEW1 as the selected view.
            DISPLAY:SELECT:VIEW? might return DISPLAY:SELECT:VIEW MATHFFT1 indicating MATHFFT1 is the selected view.

DISPLAY:SELECT:WAVEView<x>:SOURce

This command sets or queries the selected source in the given waveview.

Group          Display Control

Syntax   DISPLAY:SELECT:WAVEView<x>:SOURce
          {CH<x>|MATH<x>|BUS<x>|REF<x>|PLOT<x>}

Arguments   Arguments are the selected source.
Examples  
DISplay:SELect:WAVEView1:SOUrce MATH1 sets MATH1 as the selected source.

DISplay:SELect:WAVEView1:SOUrce? might return :DISPLAY:SELECT:WAVEVIEW1:SOURCE CH2 indicating CH2 is the selected source.

DISplay:VARpersist

This command sets or queries display persistence decay time, which is the approximate decay time for a freshly struck persistence sample.

Group  Display Control

Syntax  
DISplay:VARpersist <NR3>
DISplay:VARpersist?

Arguments  
<NR3> indicates the persistence decay time and ranges from 0.5 to 100.

Examples  
DISPlay:VARPERsIST 5 sets the persistence decay time to 5.

DISPlay:VARPERsIST? might return :DISPLAY:VARPERsIST 3.0000E-01, indicating that persistence decay time is currently set to 0.300.

DISplay:WAVEView<x>:BUS:B<x>:STATE

Sets or queries the state of the specified bus in the specified waveview.

\textbf{NOTE.} \textit{WAVEView<x> is the specified waveview and must be WAVEView1.}

Group  Display Control

Syntax  
DISplay:WAVEView<x>:BUS:B<x>:STATE \{OFF|ON|0|1\}
DISplay:WAVEView<x>:BUS:B<x>:STATE?

Arguments  
0 turns specified bus off.
1 turns the specified bus on.
Commands listed in alphabetical order

ON turns the specified bus on.
OFF turns specified bus off.

Examples

DISPLAY:WAVEView<x>:BUS:B<x>:VERTical:POStion
Sets or queries the vertical position of the specified bus in the specified waveview.

NOTE. WAVEView<x> is the specified waveview and must be WAVEView1.

Group Display Control

Syntax
DISPLAY:WAVEView<x>:BUS:B<x>:VERTical:POStion <NR3>
DISPLAY:WAVEView<x>:BUS:B<x>:VERTical:POStion?  

Arguments
<NR3> is the vertical position of the specified bus.

Examples
DISPLAY:WAVEView1:BUS:B1:VERTical:POStion 1.0e0 sets the position to 1 division.
DISPLAY:WAVEView1:BUS:B1:VERTical:POStion? might return :DISPLAY:WAVEVIEW1:BUS:B1:VERTICAL:POSITION 0.0E+0 indicating the position is 0.0 divisions.

DISPLAY:WAVEView<x>:CH<x>:STATE
Sets or queries the state of the specified channel in the specified waveview.

NOTE. WAVEView<x> is the specified waveview and must be WAVEView1.

Group Display Control

Syntax
DISPLAY:WAVEView<x>:CH<x>:STATE {<NR1>|OFF|ON}
DISPLAY:WAVEView<x>:CH<x>:STATE?
Arguments

<NR1> = 0 disables the specified channel on the specified waveview, any other value turns this feature on.
OFF disables the display the specified channel on the specified waveview.
ON enables the specified channel on the specified waveview.

Examples

DISplay:WAVEView1:CH1:STATE ON turns on channel 1.
DISplay:WAVEView1:CH1:STATE? might return :DISPLAY:WAVEVIEW1:CH1:STATE 0 indicating channel 1 is off.

DISplay:WAVEView<x>:CH<x>:VERTical:POSition

Sets or queries the vertical position of the specified channel in the specified waveview in divisions. 0.0 divisions is center, 5.0 top of the window, and -5.0 the bottom of the window.

NOTE. WAVEView<x> is the specified waveview and must be WAVEView1.

Group

Display Control

Syntax

DISplay:WAVEView<x>:CH<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:CH<x>:VERTical:POSition?

Arguments

<NR3> is the vertical position in divisions. 0.0 divisions is center, 5.0 top of the window, and -5.0 the bottom of the window.

Examples

DISplay:WAVEView1:CH1:VERTical:POSITION 1.0e0 sets the position to 1 division.

DISplay:WAVEView<x>:CH<x>:VERTical:SCAle

Sets or queries the vertical scale of the specified channel in volts per division within the specified waveview.

NOTE. WAVEView<x> is the specified waveview and must be WAVEView1.
Group: Display Control

Syntax:

```
DISPLAY:WAVEView<x>:CH<x>:VERTical:SCAle <NR3>
DISPLAY:WAVEView<x>:CH<x>:VERTical:SCAle?
```

Arguments:

- `<NR3>` is the vertical scale of the specified channel.

Examples:

```
DISPLAY:WAVEView1:CH1:VERTical:SCAle 2.0e2 sets the vertical scale to 200 mV per division.
```

**DISPLAY:WAVEView<x>:CH<x>_DALL:STATE**

This command sets or queries the display state of the specified digital channel in the specified waveview.

Group: Display Control

Syntax:

```
DISPLAY:WAVEView<x>:CH<x>_DALL:STATE {<NR1>|OFF|ON}
DISPLAY:WAVEView<x>:CH<x>_DALL:STATE?
```

Arguments:

- `<NR1>` = 0 disables the display of the specified channels on the specified waveview, any other value turns this feature on.
- OFF disables the display of the specified channels on the specified waveview.
- ON enables the display of the specified channels on the specified waveview.

Examples:

```
DISPLAY:WAVEView1:CH1_DALL:STATE ON enables the display of the specified channels on the specified waveview.
DISPLAY:WAVEView1:CH1_DALL:STATE? might return :DISPLAY:WAVEVIEW1:CH1_DALL:STATE 0 indicating specified digital channels in the specified waveview are off.
```
**DISplay:WAVEView<x>:CH<x>_DALL:VERTical:POsition**

This command sets or queries the vertical position of the specified digital channel in the specified waveview in divisions. The position ranges from 5.0 to -5.0 divisions.

**Group**
Display Control

**Syntax**

- `DISplay:WAVEView<x>:CH<x>_DALL:VERTical:Position <NR3>`
- `DISplay:WAVEView<x>:CH<x>_DALL:VERTical:Position?`

**Arguments**

- `<NR3>` is the vertical position of the specified digital channel in the specified waveview in divisions.

**Examples**

- `DISplay:WAVEView1:CH1_DALL:VERTical:Position 1.0e0` sets the vertical position to 1.0 divisions.
- `DISplay:WAVEView1:CH1_DALL:VERTical:Position?` might return `:DISPLAY:WAVEVIEW1:CH1_DALL:VERTICAL:POSITION 0.0E+0` indicating the vertical position is 0.0 divisions.

**DISplay:WAVEView<x>:CH<x>_D<x>:STATE**

This command sets or queries the display state of the specified digital channel in the specified waveview.

**Group**
Display Control

**Syntax**

- `DISplay:WAVEView<x>:CH<x>_D<x>:STATE {<NR1>|OFF|ON}`
- `DISplay:WAVEView<x>:CH<x>_D<x>:STATE?`

**Arguments**

- `<NR1> = 0` disables the display of the specified channel on the specified waveview, any other value turns this feature on.
- **OFF** disables the display of the specified channel on the specified waveview.
- **ON** enables the display of the specified channel on the specified waveview.

**Examples**

- `DISplay:WAVEView1:CH1_D1:STATE OFF` turns off the specified digital channel.
DISPLAY:WAVEview1:CH1_D1:STATE? might return
:DISPLAY:WAVEVIEW1:CH1_D1:STATE 1 indicating the specified digital channel is on.

**DISPLAY:WAVEView<x>:CURSor? (Query Only)**

This query returns the cursor parameters for the specified waveview.

**NOTE. WAVEview<x> is the specified waveview and must be WAVEview1.**

**Group** Cursor

**Syntax** DISPLAY:WAVEView<x>:CURSor?

**Returns** Returns the cursor parameters for the specified waveview.

**Examples**
DISPLAY:WAVEView1:CURSor? might return
WAVEFORM;AUTO;2.4993299784074;2.495885115371;
-3.4414668702805E-3;"A";"A";300.0E-6;2.236966999970;
256.967000E-3;-300.0E-6;-300.0E-6;300.0E-6; "????
????";"s";????  ????";600.0E-6;1.666666666667E+3;
1;300.0E-6;-300.0E-6;SAME;4.1655499640123E+3;INDEPENDENT;AUTO

**DISPLAY:WAVEView<x>:CURSor:CURSOR<x>? (Query Only)**

This query returns the cursor parameters for the specified cursor in the specified waveview.

**NOTE. WAVEview<x> is the specified waveview and must be WAVEview1.**

**Group** Cursor

**Syntax** DISPLAY:WAVEView<x>:CURSor:CURSOR<x>?

**Returns** Returns the cursor parameters for the specified cursor in the specified waveview.
Examples

DISplay:WAVEView1:CURSor:CURSOR1? might return
:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:WAVEFORM:BPOSITION
5.0E-6;APOSITION
-5.0E-6;DISPLAY:WAVEVIEW1:CURSOR:CURSOR:B SOURCE
AUTO;HBARS:BPOSITION 9.91E+37;BUNITS
"V";AUNITS "V";APOSITION 9.91E+37;DELTA
9.91E+37;DISPLAY:WAVEVIEW1:CURSOR:CURSOR:VBA
RS:BPOSITION 5.0E-6;APOSITION -5.0E-6;UNITS "s";DELTA
10.0E-6;ALTERNATEB "????????";ALTERNATEA
"????????";DISPLAY:WAVEVIEW1:CURSOR:CURSOR:STATE
1;FUNCTION WAVEFORM,SCREEN:AYPOSITION
-2.0E-3;BXPOSITION -5.0E-6;BYPOSITION -2.0E-3;AXPOSITION
-5.0E-6;DISPLAY:WAVEVIEW1:CURSOR:CURSOR:MODE
INDEPENDENT;ASOURCE AUTO;DDT
9.91E+37;ONEOVERDELTATVALUE 100.0E+3;SPLITMODE
SAME,LINESTYLE SOLID.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:ASOURce

This command sets or queries the cursor A source of the specified cursor in the specified waveview.

NOTE. WAVEView<x> is the specified waveview and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

DISplay:WAVEView<x>:CURSor:CURSOR<x>:ASOURce
{AUTO|CH<x>|BUS<x>|MATH<x>|REF<x>|PLOT<x>}?

Arguments

Arguments are the specified source waveform.

Examples

DISplay:WAVEView1:CURSor:CURSOR:ASOURce CH1 sets the cursor 1 source to channel 1.

DISplay:WAVEView1:CURSor:CURSOR:ASOURce? might return
:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:ASOURce REF1 indicating the cursor A source is reference 1.
**DISplay:WAVEView<x>:CURSor:CURSOR<x>:BSOUrce**

This command sets or queries the cursor B source of the specified cursor in the specified waveview.

**NOTE.** \( WAVEview<x> \) is the specified waveview and must be \( WAVEview1 \). \( CURSOR<x> \) is the specified cursor and must be \( CURSOR1 \).

**Group** 
Cursor

**Syntax**

\[
\text{DISplay:WAVEView}<x>:\text{CURSor}:\text{CURSOR}<x>:\text{BSOUrce} \\
\{ \text{CH}<x>|\text{BUS}<x>|\text{MATH}<x>|\text{REF}<x>|\text{PLOT}<x> \} \\
\text{DISplay:WAVEView}<x>:\text{CURSor}:\text{CURSOR}<x>:\text{BSOUrce}? 
\]

**Arguments** 
Arguments are the specified source waveform.

**Examples**

\[
\text{DISplay:WAVEView1:CURSor:CURSOR1:BSOUrce}\ CH1 \text{ sets the cursor B source to channel 1.}
\]

\[
\text{DISplay:WAVEView1:CURSor:CURSOR1:BSOUrce}? \text{ might return } :\text{DISplay:WAVEView1:CURSor:CURSOR1:BSOUrce}\ \text{REF1 indicating the cursor B source is reference 1.}
\]

**DISplay:WAVEView<x>:CURSor:CURSOR<x>:DDT? (Query Only)**

This query returns the delta V over delta T cursor readout value of the specified cursor in the specified waveview.

**NOTE.** \( WAVEview<x> \) is the specified waveview and must be \( WAVEview1 \). \( CURSOR<x> \) is the specified cursor and must be \( CURSOR1 \).

**Group** 
Cursor

**Syntax**

\[
\text{DISplay:WAVEView}<x>:\text{CURSor}:\text{CURSOR}<x>:\text{DDT}?
\]

**Returns** 
The delta V over delta T cursor readout value of the specified cursor in the specified waveview.
Examples


DISplay:WAVEView<x>:CURSor:CURSOR<x>:FUNCTION

This command sets or queries the cursor type of the specified cursor in the specified waveview.

**NOTE.** \texttt{WAVEView<x>} is the specified waveview and must be \texttt{WAVEView1}. \texttt{Cursor<x>} is the specified cursor and must be \texttt{CURSOR1}.

**Group**

Cursor

**Syntax**

DISplay:WAVEView<x>:CURSor:CURSOR<x>:FUNCTION {SCREEN|WAVEFORM|VBArs|HBArs}  
DISplay:WAVEView<x>:CURSor:CURSOR<x>:FUNCTION?

**Arguments**

HBArs specifies horizontal bar cursors, which measure in vertical units.  
VBArs specifies vertical bar cursors, which measure in horizontal units.  
SCREEN specifies both horizontal and vertical bar cursors, which measure in horizontal and vertical units specified by the Cursor 1 and Cursor 2 Sources. Use these cursors to measure anywhere in the waveform display area.  
WAVEform specifies paired or split cursors in YT display format for measuring waveform amplitude and time. In XY and XYZ format, these cursors indicate the amplitude positions of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis and Ch2 is the Y axis) relative to the trigger.

**Examples**

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION VBArs selects the vertical bar cursor type for the specified waveview and cursor.  
DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION HBArs indicating that the specified cursor is set to HBArs.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:APOSition

Sets or queries the HBARs vertical A position of the specified cursor in the specified waveview.
Commands listed in alphabetical order

**NOTE. WAVEView<x> is the specified waveview and must be WAVEView1**

**Cursor<x> is the specified cursor and must be CURSOR1.**

<table>
<thead>
<tr>
<th>Group</th>
<th>Cursor</th>
</tr>
</thead>
</table>

**Syntax**

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:APOsition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:APOsition?

**Arguments**

<NR3> is the vertical cursor A position of the specified cursor in the specified waveview. 0.0 divisions is center, 5.0 top of the waveview, and -5.0 the bottom of the waveview.

**Examples**

DISplay:WAVEView1:CURSor:CURSOR1:HBARS:APOSITION 3.0e0 sets the cursor position to 3 V.

DISplay:WAVEView1:CURSor:CURSOR1:HBARS:APOSITION? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:APOSITION 2.9303448275862 indicating the cursor position is 2.93 V.

**DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArS:AUNIts? (Query Only)**

This command queries the cursor A vertical units of the specified cursor in the specified waveview.

**NOTE. WAVEView<x> is the specified waveview and must be WAVEView1**

**Cursor<x> is the specified cursor and must be CURSOR1.**

<table>
<thead>
<tr>
<th>Group</th>
<th>Cursor</th>
</tr>
</thead>
</table>

**Syntax**

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArS:AUNIts?

**Arguments**

<QString> is the cursor A vertical units of the specified cursor in the specified waveview.

**Examples**

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BPOSition

Sets or queries the HBARs vertical B position of the specified cursor in the specified waveview.

**NOTE.** WAVEView<x> is the specified waveview and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

**Group:** Cursor

**Syntax:**

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BPOSition?

**Arguments:**

<NR3> is the vertical cursor B position of the specified cursor in the specified waveview.

**Examples:**

DISplay:WAVEView1:CURSor:CURSOR1:HBArs:BPOSition 4.0e-1 sets the position to 400 mV.

DISplay:WAVEView1:CURSor:CURSOR1:HBArs:BPOSition? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARs:BPOSITION 387.9148706896567E-3 indicating the B cursor position is 387.9 mV.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BUNIts? (Query Only)

This command queries the cursor B vertical units of the specified cursor in the specified waveview.

**NOTE.** WAVEView<x> is the specified waveview and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

**Group:** Cursor

**Syntax:**

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BUNIts?

**Returns:**

<QString> is the cursor B vertical units of the specified cursor in the specified waveview.
Examples

`DISPLAY:WAVEView1:CURSor:CURSOR1:HBAr5:BUNIts?` might return
`DISPLAY:WAVEView1:CURSOR:CURSOR1:HBARS:BUNITS "V"` indicating
the BUNITS are set to Volts.

**DISplay:WAVEView<x>:CURSoR:CURSOR<x>:HBAr5:DELTa?** (Query Only)

This command queries the delta V cursor readout value of the specified cursor in
the specified waveview.

**NOTE.** `WAVEView<x>` is the specified waveview and must be `WAVEView1`
`CURSoR<x>` is the specified cursor and must be `CURSOR1`.

**Group**

Cursor

**Syntax**

`DISPLAY:WAVEView<x>:CURSoR:CURSOR<x>:HBAr5:DELTa?`

**Returns**

The delta V cursor readout value of the specified cursor in the specified waveview.

**Examples**

`DISPLAY:WAVEView1:CURSor:CURSOR1:HBAr5:DELTa?` might return
`DISPLAY:WAVEView1:CURSOR:CURSOR1:HBARS:DELTa 2.612085129310` indicating the delta between the cursors is 2.61 V.

**DISplay:WAVEView<x>:CURSoR:CURSOR<x>:MODe**

Sets or queries the cursor tracking mode of the specified cursor in the specified
waveview.

**NOTE.** `WAVEView<x>` is the specified waveview and must be `WAVEView1`
`CURSoR<x>` is the specified cursor and must be `CURSOR1`.

**Group**

Cursor

**Syntax**

`DISPLAY:WAVEView<x>:CURSoR:CURSOR<x>:MODe`  
`{INDEPENDENT|TRACK}`

**Arguments**

`TRACK` ties the navigational functionality of the two cursors together. For cursor 1
adjustments, this ties the movement of the two cursors together, however, cursor 2
continues to move independently of cursor 1.
INDEPENDENT allows independent adjustment of the two cursors.

Examples
DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:MODE TRACK specifies that the cursor positions move in unison.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:ONEOVERDELTATVALUE? (Query Only)

This query returns the one over delta T cursor readout value of the specified cursor in the specified waveview.

NOTE. WAVEView<x> is the specified waveview and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group Cursor

Syntax DISplay:WAVEView<x>:CURSor:CURSOR<x>:ONEOVERDELTATVALUE?

Returns The one over delta T cursor readout value of the specified cursor in the specified waveview.

Examples DISplay:WAVEView1:CURSor:CURSOR1:ONEOVERDELTATVALUE? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:ONEOVERDELTATVALUE 179.3188E+3 indicating the one over delta time value is 179.3 kHz.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AXPOSition

Sets or queries the horizontal cursor A position of the specified cursor in the specified waveview.

NOTE. WAVEView<x> is the specified waveview and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group Cursor
Commands listed in alphabetical order

Syntax

`Display:WaveView<x>:Cursor:Cursor<y>:Screen:AXPosition <NR3>`

`Display:WaveView<x>:Cursor:Cursor<y>:Screen:AXPosition?`

Arguments

`<NR3>` is the horizontal cursor A position of the specified cursor in the specified waveview.

Examples

`Display:WaveView1:Cursor:Cursor1:Screen:AXPosition 15.0E-3` sets the x position of screen cursor1 in waveview1 to 15 mV.

`Display:WaveView1:Cursor:Cursor1:Screen:AXPosition?` might return:

`:Display:WaveView1:Cursor:Cursor1:Screen:AXPosition -64.0000E-03` indicating that x position of the specified screen cursor in the specified waveview is set to -64 mV.

---

`Display:WaveView<x>:Cursor:Cursor<y>:Screen:AYPosition`

This command sets or queries the vertical cursor A position of the specified cursor in the specified waveview.

**NOTE.** `WaveView<x>` is the specified waveview and must be `WaveView1`.

`Cursor<y>` is the specified cursor and must be `Cursor1`.

---

Group

Cursor

Syntax

`Display:WaveView<x>:Cursor:Cursor<y>:Screen:AYPosition <NR3>`

`Display:WaveView<x>:Cursor:Cursor<y>:Screen:AYPosition?`

Arguments

`<NR3>` the vertical cursor A position of the specified cursor in the specified waveview.

Examples

`Display:WaveView1:Cursor:Cursor1:Screen:AYPosition 25.0E-3` sets the y position of the specified screen cursor of the specified waveview to 25 mV.

`Display:WaveView1:Cursor:Cursor1:Screen:AYPosition?` might return:

`:Display:WaveView1:Cursor:Cursor1:Screen:AYPosition -53.0000E-03` indicating that y position of the specified screen cursor of the specified waveview is set to -53 mV.
**DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOsition**

Sets or queries the horizontal cursor B position of the specified cursor in the specified waveview.

**NOTE.** `WAVEView<x>` is the specified waveview and must be `WAVEView1`. `Cursor<x>` is the specified cursor and must be `CURSOR1`.

**Group**

Cursor

**Syntax**

```plaintext
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOsition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOsition?
```

**Arguments**

<NR3> is the horizontal cursor B position of the specified cursor in the specified waveview.

**Examples**

```plaintext
DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOsITION 15.0E-3 sets the x position of the specified screen cursor in the specified waveview to 15 mV.
DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOsITION? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOsITION -64.0000E-03 indicating that x position of the specified screen cursor in the specified waveview is set to -64 mV.
```

**DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOsition**

This command sets or queries the vertical cursor B position of the specified cursor in the specified waveview.

**NOTE.** `WAVEView<x>` is the specified waveview and must be `WAVEView1`. `Cursor<x>` is the specified cursor and must be `CURSOR1`.

**Group**

Cursor

**Syntax**

```plaintext
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOsition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOsition?
```

**Arguments**

<NR3> the vertical cursor B position of the specified cursor in the specified waveview.
Examples

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:BYPOSITION 25.0E-3 sets the y position of the screen cursor of waveview1 to 25 mV.


DISplay:WAVEView<x>:CURSor:CURSOR<x>:SPLITMODE

This command sets or queries whether both cursors have the same or different sources.

NOTE. WAVEView<x> is the specified waveview and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group  Cursor

Syntax  DISplay:WAVEView<x>:CURSor:CURSOR<x>:SPLITMODE {SAME|SPLIT}
        DISplay:WAVEView<x>:CURSor:CURSOR<x>:SPLITMODE?

Arguments  SAME specifies both cursors have the same source.
            SPLIT specifies the cursors have different sources.

Examples  DISplay:WAVEView1:CURSor:CURSOR1:SPLITMODE SPLIT specifies the cursors have different sources.

DISplay:WAVEView1:CURSor:CURSOR1:SPLITMODE? might return :DISplay:WAVEView1:CURSor:CURSOR1:SPLITMODE SAME indicating the cursors have the same source.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:STATE

This command sets or queries the visible state of the specified cursor in the specified waveview.

NOTE. WAVEView<x> is the specified waveview and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group  Cursor
Syntax

DISplay:WAVEView<x>:CURSor:CURSOR<x>:STATE {<NR1>|OFF|ON}
DISplay:WAVEView<x>:CURSor:CURSOR<x>:STATE?

Arguments

<NR1> = 0 disables the specified cursor in the specified waveview, any other value turns this feature on.

OFF disables the specified cursor in the specified waveview.

ON enables the specified cursor in the specified waveview.

Examples

DISplay:WAVEView1:CURSor:CURSOR1:STATE OFF turns the specified cursor off.

DISplay:WAVEView1:CURSor:CURSOR1:STATE? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:STATE 1 indicating the specified cursor in the specified waveview is on.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:APOSition

This command sets or queries the cursor A horizontal position of the specified cursor in the specified waveview.

NOTE. WAVEView<x> is the specified waveview and must be WAVEView1.
CURSor<x> is the specified cursor and must be CURSOR1.

Group

Cursor

Syntax

DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:APOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:APOSition?

Arguments

<NR3> is the horizontal cursor A position of the specified cursor in the specified waveview.

Examples

DISplay:WAVEView1:CURSor:CURSOR1:VBArs:APOSition -4.0e-6 set the cursor position to 4.0 μs.

DISplay:WAVEView1:CURSor:CURSOR1:VBArs:APOSition? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:VBArs:APOSITION -5.0E-6 indicating the cursor A position is -5.0 μs.
Commands listed in alphabetical order

**DISplay:WAVEView<x>:CURSOR:CURSOR<x>:VBAr:s:BPOSITION**

This command sets or queries the cursor B horizontal position of the specified cursor in the specified waveview.

**NOTE.** *WAVEView<x>* is the specified waveview and must be *WAVEView1*.  
*CURSOR<x>* is the specified cursor and must be *CURSOR1*.

**Group**  
Cursor

**Syntax**  
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBAr:s:BPOSITION <NR3>  
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBAr:s:BPOSITION?

**Arguments**  
<NR3> is the horizontal cursor B position of the specified cursor in the specified waveview.

**Examples**  
DISplay:WAVEView1:CURSor:CURSOR1:VBAr:s:BPOSITION 4.0e-6 sets the cursor position to 4.0 μs.  

**DISplay:WAVEView<x>:CURSOR:CURSOR<x>:VBAr:s:DELTa? (Query Only)**

This query sets or returns the delta T cursor readout value of the specified cursor in the specified waveview.

**NOTE.** *WAVEView<x>* is the specified waveview and must be *WAVEView1*.  
*CURSOR<x>* is the specified cursor and must be *CURSOR1*.

**Group**  
Cursor

**Syntax**  
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBAr:s:DELTa?

**Returns**  
The delta T cursor readout value of the specified cursor in the specified waveview.
Examples  

**DISplay:**WAVEView<x>**:CURSor:**CURSOR<x>**:VBArS:**UNIts? (Query Only)**

This query returns cursor A vertical units of the specified cursor in the specified waveview.

**NOTE.**  
WAVEView<x> is the specified waveview and must be WAVEView1.  
CURSOR<x> is the specified cursor and must be CURSOR1.

**Group**  
Cursor

**Syntax**  
DISplay:WAVEView<x>**:CURSor:**CURSOR<x>**:VBArS:**UNIts?

**Returns**  
The cursor A vertical units of the specified cursor in the specified waveview.

**Examples**  
DISplay:WAVEView1:CURSor:CURSOR1:VBArS:**UNIts?  might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:VBARS:UNITS "s" indicating the cursor units are seconds.

**DISplay:**WAVEView<x>**:CURSor:**CURSOR<x>**:WAVEform:**APOSition**

This command sets or queries the horizontal cursor A position of the specified cursor in the specified waveview.

**NOTE.**  
WAVEView<x> is the specified waveview and must be WAVEView1.  
CURSOR<x> is the specified cursor and must be CURSOR1.

**Group**  
Cursor

**Syntax**  
DISplay:WAVEView<x>**:CURSor:**CURSOR<x>**:WAVEform:**APOSition <NR3>  
DISplay:WAVEView<x>**:CURSor:**CURSOR<x>**:WAVEform:**APOSition?

**Arguments**  
<NR3> is the horizontal cursor A position of the specified cursor in the specified waveview.
Examples

DISplay:WAVEView1:CURSor:CURSOR1:WAVEform:APOSition 1.5e-6
sets the position to 1.5 μs.

DISplay:WAVEView1:CURSor:CURSOR1:WAVEform:APOSition? might
return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:WAVEFORM:APOSITION
-1.2667480236557E-6 indicating the position is 1.27 μs.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:BPOSition

This command sets or queries the horizontal cursor B position of the speci
fied cursor in the specified waveview.

**NOTE.** WAVEView<x> is the specified waveview and must be WAVEView1
Cursor<x> is the specified cursor and must be CURSOR1.

Group  Cursor

Syntax

DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:BPOSition
<NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:BPOSition?

Arguments

<NR3> is the horizontal cursor B position of the specified cursor in the specified
waveview.

Examples

DISplay:WAVEView1:CURSor:CURSOR1:WAVEform:BPOSition 8.0e-9
sets the position to 8.0 ns.

DISplay:WAVEView1:CURSor:CURSOR1:WAVEform:BPOSition? might
return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:WAVEFORM:BPOSITION
711.6310718892624E-9 indicating the cursor position is 711.6 ns.

DISplay:WAVEView<x>:FILTer

This command sets or queries the type of interpolation filter for the display.

**NOTE.** WAVEView<x> is the specified waveview and must be WAVEView1

Group  Display Control

Syntax

DISplay:WAVEView<x>:FILTer {SINX|LINear}
Arguments

**LINEAR** specifies linear interpolation, where acquired points are connected with straight lines.

**SINX** specifies \( \sin(x)/x \) interpolation, where acquired points are fit to a curve.

Examples

**DISplay:**WAVEView1:**FILTER** **SINX** specifies sine-curve interpolation, when magnifying waveforms.

**DISplay:**WAVEView1:**FILTER**? might return **DISplay:**WAVEView1:**FILTER** **LINEAR**, indicating that straight-line interpolation is specified for magnifying waveforms.

**DISplay:**WAVEView<x>:**GRAticule**

This command selects or queries the type of graticule that is displayed.

**NOTE.** **WAVEView<x>** is the specified waveview and must be **WAVEView1**.

Group

Display Control

Syntax

**DISplay:**WAVEView<x>:**GRAticule** {GRID|TIME|FULL|NONE}

**DISplay:**WAVEView<x>:**GRAticule**?

Arguments

**GRID** specifies a frame and grid only.

**TIME** specifies a time graticule only.

**FULL** specifies a frame, a grid and cross hairs.

**NONE** specified no graticule.

Examples

**DISplay:**WAVEView1:**GRAticule** **TIME** specifies a time graticule.

**DISplay:**WAVEView1:**GRAticule**? might return **DISplay:**WAVEView1:**GRAticule** **GRID** indicating the graticule is a grid.

**DISplay:**WAVEView<x>:**INTENSITY:**GRAticule**

This command sets or queries the graticule saturation level.

**NOTE.** **WAVEView<x>** is the specified waveview and must be **WAVEView1**.
Group                  Display Control

Syntax                 DISplay:WAVEView<x>:INTENSITY:GRAticule <NR2>
                       DISplay:WAVEView<x>:INTENSITY:GRAticule?

Arguments              <NR2> is the graticule saturation level.

Examples               DISplay:WAVEView1:INTENSITY:GRAticule 75 sets the saturation level to 75%.
                       DISplay:WAVEView1:INTENSITY:GRAticule? might return
                       :DISPLAY:WAVEVIEW1:INTENSITY:GRATICULE 66.0000 indicating the
                       saturation level is at 66%.

DISplay:WAVEView<x>:INTENSITy:WAVEform

This command sets or queries the waveform saturation level.

NOTE. WAVEview<x> is the specified waveview and must be WAVEview1

Group                  Display Control

Syntax                 DISplay:WAVEView<x>:INTENSITY:WAVEform <NR2>
                       DISplay:WAVEView<x>:INTENSITY:WAVEform?

Arguments              <NR2> is the waveform saturation level.

Examples               DISplay:WAVEView1:INTENSITY:WAVEform 75 sets the saturation level to 75%.
                       DISplay:WAVEView1:INTENSITY:WAVEform? might return
                       :DISPLAY:WAVEVIEW1:INTENSITY:WAVEFORM 62.0000 indicating the
                       saturation level is 62%.

DISplay:WAVEView<x>:MATH:MATH<x>:AUTOScale

This command sets or queries whether the specified math gets auto-scaled when
the math equation changes within the specified waveview.
Group: Display Control

Syntax:
DISPlay:WAVEView<x>:MATH:MATH<x>:AUTOScale {<NR1>|OFF|ON}
DISPlay:WAVEView<x>:MATH:MATH<x>:AUTOScale?

Arguments:
- `<NR1>` = 0 disables the autoscaling the math in the specified waveview; any other value turns this feature on.
- OFF disables the autoscaling the math in the specified waveview.
- ON enables the autoscaling the math in the specified waveview.

Examples:
- DISPlay:WAVEView1:MATH:MATH3:AUTOScale 1 enables the autoscaling the math in the specified waveview.

DISplay:WAVEView<x>:MATH:MATH<x>:STATE

This command sets or queries the state of the specified math waveform in the specified waveview.

**NOTE.** WAVEView<x> is the specified waveview and must be WAVEView1.

Group: Display Control

Syntax:
DISPlay:WAVEView<x>:MATH:MATH<x>:STATE {<NR1>|OFF|ON}
DISPlay:WAVEView<x>:MATH:MATH<x>:STATE?

Arguments:
- `<NR1>` = 0 disables the specified math in the specified waveview; any other value turns this feature on.
- OFF disables the specified math in the specified waveview.
- ON enables the specified math in the specified waveview.

Examples:
- DISPlay:WAVEView1:MATH:MATH1:STATE OFF disables the specified math in the specified waveview.
**GROUP: Display Control**

**Syntax**: `DISPLAY:WAVEView<x>:MATH:MATH<x>:VERTical:POSition <NR3>`

This command sets or queries the vertical position in divisions of the specified math waveform in the specified waveview.

**Arguments**: `<NR3>` is the vertical position in divisions of the specified math waveform.

**Examples**
- `DISPLAY:WAVEView1:MATH:MATH1:VERTical:POSition 0` sets the position to 0 divisions.

**Syntax**: `DISPLAY:WAVEView<x>:MATH:MATH<x>:VERTical:SCAle <NR3>`

Sets or queries the vertical scale of the specified math in volts per division within the specified waveview.

**Arguments**: `<NR3>` is the vertical scale of the specified math waveform.

**Examples**
- `DISPLAY:WAVEView1:MATH:MATH1:VERTical:SCAle 0` sets the vertical scale to 0.
- `DISPLAY:WAVEView1:MATH:MATH1:VERTical:SCAle?` might return `:DISPLAY:WAVEVIEW1:MATH:MATH1:VERTICAL:SCALE 0.5` indicating the vertical scale is 0.5.
**Examples**

`DISPLAY:WAVEview1:MATH:MATH1:VERTical:SCALE 400e-3` sets the scale to 400 mV per division.

`DISPLAY:WAVEview1:MATH:MATH1:VERTical:SCALE?` might return

:DISPLAY:WAVEview1:MATH:MATH1:VERTICAL:SCALE 395.0000E-3

indicating the scale is 395 mV.

**DISPLAY:WAVEView<x>:PLOT:PLOT<x>:AUTOScale**

This command sets or queries whether the specified trend gets auto-scaled when the new data is available within the specified waveview.

**Group**

Display Control

**Syntax**

`DISPLAY:WAVEView<x>:PLOT:PLOT<x>:AUTOScale {<NR1>|OFF|ON}`

`DISPLAY:WAVEView<x>:PLOT:PLOT<x>:AUTOScale?`

**Arguments**

`<NR1> = 0` disables auto-scaling the specified plot in the specified waveview, any other value turns this feature on.

`OFF` disables auto-scaling the specified plot in the specified waveview.

`ON` enables auto-scaling the specified plot in the specified waveview.

**Examples**

`DISPLAY:WAVEview1:PLOT:PLOT1:AUTOScale 1` enables auto-scaling the specified plot in the specified waveview.

`DISPLAY:WAVEview1:PLOT:PLOT1:AUTOScale?` might return

:DISPLAY:WAVEview1:PLOT:PLOT1:AUTOSCALE 0 indicating that auto-scaling for plot 1 is off.

**DISPLAY:WAVEView<x>:PLOT:PLOT<x>:STATE**

This command sets or queries the state of the specified time trend plot waveform in the specified waveview.

**Group**

Display Control

**Syntax**

`DISPLAY:WAVEView<x>:PLOT:PLOT<x>:STATE {<NR1>|OFF|ON}`

`DISPLAY:WAVEView<x>:PLOT:PLOT<x>:STATE?`
Arguments

<NR1> = 0 disables the specified plot in the specified waveview, any other value turns this feature on.
OFF disables the specified plot in the specified waveview
ON enables the specified plot in the specified waveview.

Examples

DISPLAY:WAVEView1:PLOT:STATE OFF disables the specified plot in the specified waveview.

DISPLAY:WAVEView1:PLOT:STATE? might return
:DISPLAY:WAVEVIEW1:PLOT:STATE 1 indicating the specified plot in the specified waveview is on.

DISPLAY:WAVEView<x>:PLOT:PLOT<x>:VERTical:POSition

This command sets or queries the vertical position of the specified time trend in the specified waveview in absolute units.

Group Display Control

Syntax

DISPLAY:WAVEView<x>:PLOT:PLOT<x>:VERTical:POSITION <NR3>
DISPLAY:WAVEView<x>:PLOT:PLOT<x>:VERTical:POSITION?

Arguments

<NR3> is the vertical position.

Examples

DISPLAY:WAVEView1:PLOT:PLOT1:VERTical:POSITION 2.0e0 sets the position to 2.0 units.

DISPLAY:WAVEView1:PLOT:PLOT1:VERTical:POSITION? might return
:DISPLAY:WAVEVIEW1:PLOT:PLOT1:VERTICAL:POSITION 1.9918 indicating the position is 1.99 units.

DISPLAY:WAVEView<x>:PLOT:PLOT<x>:VERTical:SCAle

This command sets or queries the vertical scale of the specified time trend in units per division in the specified waveview.

Group Display Control

Syntax

DISPLAY:WAVEView<x>:PLOT:PLOT<x>:VERTical:SCALE <NR3>
DISPLAY:WAVEView<x>:PLOT:PLOT<x>:VERTical:SCALE?
Arguments  

<NR3> is the vertical scale.

Examples  

DISplay:WAVEView1:PLOT:PLOT1:VERTical:SCALE 725.0e-3 sets the scale to 725 mUnits.

DISplay:WAVEView1:PLOT:PLOT1:VERTical:SCALE? might return :DISPLAY:WAVEVIEW1:PLOT:PLOT1:VERTICAL:SCALE 730.0000E-3 indicating the scale is 730.0 mUnits.

### DISplay:WAVEView<x>:REF:REF<x>:STATE

This command sets or queries the state of the specified reference waveform in the specified waveview.

| NOTE | WAVEView<x> is the specified waveview and must be WAVEView1.
|

Group  

Display Control

Syntax  

DISplay:WAVEView<x>:REF:REF<x>:STATE {<NR1>|OFF|ON}  
DISplay:WAVEView<x>:REF:REF<x>:STATE?

Arguments  

<NR1> = 0 disables the specified reference in the specified waveview, any other value turns this feature on.

OFF disables the specified reference in the specified waveview.

ON enables the specified reference in the specified waveview.

Examples  

DISplay:WAVEView1:REF:REF1:STATE OFF disables the specified reference in the specified waveview.


### DISplay:WAVEView<x>:REF:REF<x>:VERTical:POSition

This command sets or queries the vertical position in divisions of the specified reference in the specified waveview.

| NOTE | WAVEView<x> is the specified waveview and must be WAVEView1.
|
Group: Display Control

Syntax:

- `DISplay:WAVEView<x>:REF:REF<x>:VERTical:POSition <NR3>`
- `DISplay:WAVEView<x>:REF:REF<x>:VERTical:POSition?`

Arguments:

- `<NR3>` is the vertical position in divisions.

Examples:

- `DISplay:WAVEView1:REF:REF1:VERTical:POSition -2.5e0` sets the position to -2.5 divisions.

**DISplay:WAVEView<x>:REF:REF<x>:VERTical:SCAle**

This command sets or queries the vertical scale of the specified reference in volts per div within the specified waveview.

**Note.** `WAVEview<x>` is the specified waveview and must be `WAVEview1`.

Group: Display Control

Syntax:

- `DISplay:WAVEView<x>:REF:REF<x>:VERTical:SCAle <NR3>`
- `DISplay:WAVEView<x>:REF:REF<x>:VERTical:SCAle?`

Arguments:

- `<NR3>` is the vertical scale of the specified reference waveform.

Examples:

- `DISplay:WAVEView1:REF:REF1:VERTical:SCAle 400e-3` sets the scale to 400 mV per division.

**DISplay:WAVEView<x>:STYle**

This command sets or queries how the waveforms are displayed for analysis mode.

**Note.** `WAVEview<x>` is the specified waveview and must be `WAVEview1.`
GROUP: Display Control

DISPlay:WAVEView<x>:STYle {VECTors|DOTsonly}

Syntax

Arguments

DOTs displays individual data points. New points immediately replace old ones.

VECTors connects adjacent data points. New points immediately replace old ones.

Examples

DISplay:WAVEView1:STYle VECTORS sets the display to connect adjacent data points.

DISplay:WAVEView1:STYle? might return :DISplay:WAVEView1:STYle DOTs, indicating that data points are not connected.

DISplay:WAVEView<x>:VIEWStyle

The command sets or queries the waveform layout style used by the display.

NOTE. WAVEView<x> is the specified waveview and must be WAVEView1.

DISplay:WAVEView<x>:VIEWStyle {OVErlay|STAcked}

Syntax

Arguments

OVErlay specifies that the display view style used by the specified waveview is overlay.

STAcked specifies that the display view style used by the specified waveview is stacked.

Examples

DISplay:WAVEView1:VIEWStyle OVERLAY sets the view style to overlay.

DISplay:WAVEView1:VIEWStyle? might return :DISPLAY:WAVEVIEW1:VIEWSTYLE STACKED indicating the view style is stacked.

DISplay:WAVEView<x>:Zoom? (Query Only)

This query returns the zoom parameters of the specified waveview.
NOTE. \textit{WAVEView<x>} is the specified waveview and must be \textit{WAVEView1}.

Group Zoom

Syntax \texttt{DISplay:WAVEView<x>:Zoom?}

Returns Returns the zoom parameters of the specified waveview.

Examples \texttt{DISplay:WAVEView1:Zoom?} might return :
\texttt{DISplay:WAVEView1:ZOOM:ZOOM1:HORIZONTAL:SCALE 2.5000;POSITION 70.0000;WINSCALE 400.0000E-9;} \texttt{DISplay:WAVEView1:ZOOM:ZOOM 1:VERTICAL:SCALE 3.8000;POSITION 2.5951;} \texttt{DISplay:WAVEView1:ZOOM:ZOOM1:STATE 0}.

\textbf{DISplay:WAVEView<x>:ZOOM:ZOOM<x>?} (Query Only)

This query returns the zoom parameters of the specified zoom in the specified waveview. \textit{x} must be 1.

NOTE. \textit{WAVEView<x>} is the specified waveview and must be \textit{WAVEView1}. \textit{ZOOM<x>} is the specified zoom and must be \textit{ZOOM1}.

Group Zoom

Syntax \texttt{DISplay:WAVEView<x>:ZOOM:ZOOM<x>?}

Returns Returns the zoom parameters of the specified zoom in the specified waveview.

Examples \texttt{DISplay:WAVEView1:ZOOM:ZOOM1?} might return :
\texttt{DISplay:WAVEView1:ZOOM:ZOOM1:HORIZONTAL:SCALE 2.5000;POSITION 70.0000;WINSCALE 400.0000E-9;} \texttt{DISplay:WAVEView1:ZOOM:ZOOM 1:VERTICAL:SCALE 3.8000;POSITION 2.5951;} \texttt{DISplay:WAVEView1:ZOOM:ZOOM1:STATE 0}.

\textbf{DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:POSition}

Sets or queries the horizontal zoom position (of the specified zoom in the specified waveview) of the zoomed waveform or zoom waveform in the display, around
which the zoom waveform displays. It is freely movable around the acquisition settings (horizontal span). An acquired waveform or reference could extend off screen. The valid zoom area does not care about the waveform itself, only the user setting for acquisition.

For example, if horizontal scale is set to 1 second, position to 50, then the acquisition area will go from -5 s to +5 s. Zoom window 0 will focus on -5 s and zoom area 100 will focus on +5 s. If the instrument is stopped and the scale changed to 0.5 s, there will be data off the ends of the display. However, 0% zoom will put the user focus on -2.5 s, the lower bound of the acquisition span.

**NOTE.** \texttt{WAVEView<x>} is the specified waveview and must be \texttt{WAVEView1}. \texttt{ZOOM<x>} is the specified zoom and must be \texttt{ZOOM1}.

**Group**

**Zoom**

**Syntax**

\texttt{DISPLAY:WAVEView<x>:ZOOM:ZOOM<x>:HORIZONTAL:POSITION <NR3>}

\texttt{DISPLAY:WAVEView<x>:ZOOM:ZOOM<x>:HORIZONTAL:POSITION?}

**Arguments**

<NR3> is a value from 0 to 100.00 and is the percent of the waveform that is to the left of screen center, when the zoom factor is 2× or greater.

**Examples**

\texttt{DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:POSITION 50} sets the horizontal position of the zoom box of waveview 1 to 50 so that it is centered horizontally on the screen.

\texttt{DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:POSITION?} might return \texttt{DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:POSITION 10.0000}, indicating that the horizontal position of the zoom box of waveview 1 is centered over the first major graticule division.

**\texttt{DISPLAY:WAVEView<x>:ZOOM:ZOOM<x>:HORIZONTAL:SCALE}**

This command sets or queries the horizontal zoom factor of the specified zoom in the specified waveview.

**NOTE.** \texttt{WAVEView<x>} is the specified waveview and must be \texttt{WAVEView1}. \texttt{ZOOM<x>} is the specified zoom and must be \texttt{ZOOM1}.

**Group**

**Zoom**
Commands listed in alphabetical order

Syntax

- `DISPLAY:WAVEView<x>:ZOOM:ZOOM<x>:HORIZONTAL:SCALE <NR3>`
- `DISPLAY:WAVEView<x>:ZOOM:ZOOM<x>:HORIZONTAL:SCALE?`

Arguments

- `<NR3>` is the amount of expansion in the horizontal direction in 1-2-4 increments of the specified zoom in the specified waveview.

Examples

- `DISPLAY:WAVEView1:ZOOM:ZOOM1:HORIZONTAL:SCALE 5` sets the horizontal zoom factor of zoom1 in waveview1 to 5x.

**DISPLAY:WAVEView<x>:ZOOM:ZOOM<x>:HORIZONTAL:WINSCALE**

This command sets or queries the overview window horizontal scale in the specified waveview.

**NOTE.** `WAVEView<x>` is the specified waveview and must be `WAVEView1`. `ZOOM<x>` is the specified zoom and must be `ZOOM1`.

Group

- `Zoom`

Syntax

- `DISPLAY:WAVEView<x>:ZOOM:ZOOM<x>:HORIZONTAL:WINSCALE <NR3>`
- `DISPLAY:WAVEView<x>:ZOOM:ZOOM<x>:HORIZONTAL:WINSCALE?`

Arguments

- `<NR3>` is the horizontal scale of the zoom window.

Examples


**DISPLAY:WAVEView<x>:ZOOM:ZOOM<x>:STATE**

This command sets or queries the zoom display state of the specified zoom in the specified waveview. This command is equivalent to pushing the zoom button on the front panel.

**NOTE.** `WAVEView<x>` is the specified waveview and must be `WAVEView1`. `ZOOM<x>` is the specified zoom and must be `ZOOM1`. 
Group  
Zoom

Syntax  
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:STATE {ON|OFF|<NR1>}
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:STATE?

Arguments  
ON turns the specified zoom on.
OFF turns specified zoom off.

<NR1> = 0 disables the specified zoom; any other value enables the specified zoom.

Examples  
DISplay:WAVEView1:ZOOM:ZOOM1:STATE ON turns the specified zoom on.
DISplay:WAVEView1:ZOOM:ZOOM1:STATE? might return
:DISplay:WAVEView1:ZOOM:ZOOM1:STATE 1, indicating that specified zoom is on.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POsition

This command sets or queries the vertical position of the specified zoom in the specified waveview. It is freely movable within the confines of the acquired waveform. It is measured from the top to bottom of the acquisition window. The top of the zoom window is -5 * vertical zoom factor. The bottom of the zoom window is +5 * the vertical zoom factor. For a zoom of 5x, the position ranges from -25 to 25.

NOTE. WAVEView<x> is the specified waveview and must be WAVEView1.
ZOOM<x> is the specified zoom and must be ZOOM1.

Group  
Zoom

Syntax  
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POsition <NR3>
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POsition?

Arguments  
NR3 is the vertical position of the specified zoom in the specified waveview. It is freely movable within the confines of the acquired waveform. The top of the zoom window is -5 * vertical zoom factor. The bottom of the zoom window is +5 * the vertical zoom factor. For a vertical zoom of 5x, the position ranges from -25 to 25.

Examples  
DISplay:WAVEView1:ZOOM:ZOOM1:VERTical:POSition 50 sets the vertical position of the specified zoom in the specified waveview to 50, where the vertical
Commands listed in alphabetical order

Zoom factor is 10x, top of screen is -50 and bottom if +50, and 0 is the vertical center.

`DISPLAY:WAVEView1:ZOOM:ZOOM1:VERTical:POSITION?` might return `:DISPLAY:WAVEView1:ZOOM:ZOOM1:VERTical:POSITION 23.90000`, indicating that the vertical position of the specified zoom in the specified waveview is set to 23.9, where the vertical zoom factor is 10x, top of screen is -50 and bottom if +50, and 0 is the vertical center.

**DISPLAY:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALE**

This command sets or queries the vertical zoom factor of the specified zoom in the specified waveview.

**NOTE.** `WAVEView<x>` is the specified waveview and must be `WAVEView1`. `ZOOM<x>` is the specified zoom and must be `ZOOM1`.

**Group**  
Zoom

**Syntax**

`DISPLAY:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALE <NR3>`  
`DISPLAY:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALE?`

**Arguments**

<NR3> is the amount of vertical expansion or compression. Based on the value that you entered, this command uses the nearest scale factor. Setting the vertical scale to 1 indicates unity (no zoom).

**Examples**

`DISPLAY:WAVEView1:ZOOM:ZOOM2:VERTical:SCALE 5` sets the vertical scale of zoom1 of waveview1 to 5x.


**DISPLAY:WAVEView<y>:REF:REF<x>:FRAME**

This command sets or returns the selected frame of the specified analog ref. Each ref has a unique selected frame.

**Group**  
Display Control
Syntax

DISplay:WAVEview<y>:REF:REF<x>:FRAME <NR1>
DISplay:WAVEview<y>:REF:REF<x>:FRAME?

Arguments

<NR1> is the selected frame of the specified analog ref.

Examples

DISplay:WAVEview1:REF:REF2:FRAME? might return
:DISPLAY:WAVEVIEW1:REF:REF2:FRAME 1, indicating the selected frame is 1.

DISplay:WAVEView<y>:REF<x>_DALL:FRAME

This command sets or returns the selected frame of the specified digital ref. Each ref has a unique selected frame.

Group

Display Control

Syntax

DISplay:WAVEview<y>:REF<x>_DALL:FRAME <NR1>
DISplay:WAVEview<y>:REF<x>_DALL:FRAME?

Arguments

<NR1> is the selected frame of the specified digital ref.

Examples

DISplay:WAVEview<y>:REF<x>_DALL:FRAME? might return
:DISPLAY:WAVEVIEW1:REF1_DALL:FRAME 1, indicating the selected frame is 1.

DVM (No Query Form)

Resets the Digital Voltmeter measurements and history.

Conditions

Requires DVM option (free with product registration).

Group

DVM

Syntax

DVM RESET

Arguments

RESET specifies resetting DVM measurements and history.
**Examples**

DVM:RESET resets the DVM measurement and history.

**DVM:AUTORange**

Sets (or queries) the autorange state for the Digital Voltmeter.

*NOTE.* the DVM will not autorange as long as the DVM source is the same channel as the trigger source.

**Conditions**

Requires DVM option (free with product registration).

**Group**

DVM

**Syntax**

DVM:AUTORange {0|1|OFF|ON}

DVM:AUTORange?

**Arguments**

1 or ON turns on autorange for the Digital Voltmeter.

0 or OFF turns autorange off.

**Examples**

DVM:AUTOR ON turns on autorange for the Digital Voltmeter.

**DVM:MEASUrement:FREQuency? (Query Only)**

This command returns the current frequency value for the DVM.

**Conditions**

Requires DVM option (free with product registration).

**Group**

DVM

**Syntax**

DVM:MEASUrement:FREQuency?

**Related Commands**

DVM:TRIGger:FREQuency:COUNTer

**Examples**

DVM:MEASU:FREQ? might return 100.0000E+3, which represents 100 kHz as the current frequency value for the DVM.
DVM:MEASUrement:HIStory:AVErage? (Query Only)

Returns the average DVM readout value over the history period. The history period is a constant period of 5 seconds.

Conditions
Requires DVM option (free with product registration).

Group
DVM

Syntax
DVM:MEASUrement:HIStory:AVErage?

Examples

DVM:MEASUrement:HIStory:MAXimum? (Query Only)

Returns the maximum readout value for the DVM function over the history period. The history period is a constant period of 5 seconds.

Conditions
Requires DVM option (free with product registration).

Group
DVM

Syntax
DVM:MEASUrement:HIStory:MAXimum?

Examples

DVM:MEASUrement:HIStory:MINimum? (Query Only)

Returns the minimum readout value for the DVM over the history period. The history period is a constant period of 5 seconds.

Conditions
Requires DVM option (free with product registration).

Group
DVM
DVM:MEASUrement:HIStory:MINImum? (Query Only)

Returns the minimum readout value for the DVM function over the history period in volts.

Syntax
DVM:MEASU:HI:MINI?

Examples
DVM:MEASU:HI:MINI? might return :DVM:MEASUREMENT:HIStory:MINIMUM 430.9000E-3, which represents the minimum readout value for the DVM function over the history period in volts.

DVM:MEASUrement:INFMAXimum? (Query Only)

Returns the maximum DVM readout value over the entire time that the DVM has been on since the last change using the DVM:MODe or DVM:SOUrce commands or DVM RESET.

Syntax
DVM:MEASU:INFMAX?

Examples
DVM:MEASU:INFMAX? might return :DVM:MEASUREMENT:INFMAXIMUM 432.9000E-3, which represents the maximum readout value (in volts) of the DVM function over the entire time that the DVM has been on since the last change using the DVM:MODe or DVM:SOUrce commands or DVM RESET.

DVM:MEASUrement:INFMINimum? (Query Only)

Returns the minimum readout value of the DVM over the entire time that the DVM has been on since the last change using the DVM:MODe or DVM:SOUrce commands or DVM RESET.

Syntax
DVM:MEASU:INFMIN?

Examples
DVM:MEASU:INFMIN? might return :DVM:MEASUREMENT:INFMINIMUM 427.3000E-3, which represents the minimum readout value of the DVM.
function (in volts) over the entire time that the DVM has been on since the last change using the DVM:MODE or DVM:SOURce commands or DVM:RESET.

DVM:MEASUrement:VALue? (Query Only)

Returns the DVM readout value (the largest displayed value at the top of the DVM screen).

**Conditions**
Requires DVM option (free with product registration).

**Group**
DVM

**Syntax**
DVM:MEASUREMENT:VALUE?

**Examples**
DVM:MEASU:VAL? might return :DVM:MEASUREMENT:VALUE 430.7000E-3, which represents the DVM value.

DVM:MODe

This command specifies (or queries) the mode to use for the Digital Voltmeter.

**Conditions**
Requires DVM option (free with product registration).

**Group**
DVM

**Syntax**
DVM:MODe {ACRMS | ACDCRMS | DC | OFF}
DVM:MODe?

**Arguments**
ACRMS – displays the root-mean-square value of the acquired data, with the DC component removed.
ACDCRMS – displays the RMS value of the acquired data.
DC – displays the DC value of the acquired data.
OFF

**Examples**
DVM:MODE DC sets the mode for the DVM to DC, which displays the DC value of the acquired data.
DVM:MOD? might return ACRMS, which indicates the mode is currently set to ACRMS.

### DVM:SOURce

This command sets (or queries) the source for the DVM.

**Conditions**
Requires DVM option (free with product registration).

**Group**
DVM

**Syntax**

```
DVM:SOURce {CH<x>}
DVM:SOURce?
```

**Arguments**

- `CH<x>` specify which channel to use as the source for the DVM.

**Examples**

```
DVM:SOURce CH4 sets the DVM source to Channel 4.
```

### DVM:TRIGger:FREQuency:COUNTer

This command sets or queries the state of the trigger frequency counter readout in the trigger badge.

**Conditions**
Requires DVM option (free with product registration).

**Group**
DVM

**Syntax**

```
DVM:TRIGger:FREQuency:COUNTer {0|1|OFF|ON}
DVM:TRIGger:FREQuency:COUNTer?
```

**Arguments**

- `1` or `ON` turns on the trigger frequency counter for the Digital Voltmeter.
- `0` or `OFF` turns it off.

**Examples**

```
DVM:TRIGGER:FREQUENCY:COUNTER 0 turns off the counter.
```
*ESE

This command sets and queries the bits in the Event Status Enable Register (ESER). The ESER prevents events from being reported to the Status Byte Register (STB). For a more detailed discussion of the use of these registers, see Registers.

**Group**
Status and Error

**Syntax**

*ESE <NR1>
*ESE?

**Related Commands**

*CLS
DESE
*ESR?
EVENT?
EVMsg?
*SRE
*STB?

**Arguments**

<NR1> specifies the binary bits of the ESER according to this value, which ranges from 0 through 255.

The power-on default for the ESER is 0 if *PSC is 1. If *PSC is 0, the ESER maintains the previous power cycle value through the current power cycle.

**NOTE.** Setting the DESER and the ESER to the same values allows only those codes to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the Status Byte Register. Use the DESE command to set the DESER.

**Examples**

*ESE 209 sets the ESER to binary 11010001, which enables the PON, URQ, EXE, and OPC bits.

*ESE? might return 186, showing that the ESER contains DESE the binary value 1011010.
*ESR? (Query Only)

This query-only command returns the contents of the Standard Event Status Register (SESR). *ESR? also clears the SESR (since reading the SESR clears it). For a more detailed discussion of the use of these registers, see Registers.

**Group**
Status and Error

**Syntax**
*ESR?

**Related Commands**
ALLEv?
*CLS
DESE
*ESE
EVENT?
EVMsg?
*SRE
*STB?

**Examples**
*ESR? might return *ESR 213, showing that the SESR contains the binary value 11010101.

ETHERnet:DHCPbootp

This command sets the network configuration method to DHCP (that is ON) or static IP address (that is OFF).

**Group**
Ethernet

**Syntax**
ETHERnet:DHCPbootp {ON|OFF}
ETHERnet:DHCPbootp?

**Arguments**
ON enables the oscilloscope to search the network for a DHCP server in order to automatically assign a dynamic IP address to the oscilloscope.
NOTE. Do not use DHCP searching if your oscilloscope has been assigned a static address on a network. If you set this command to ON, the DHCP search will delete or change your static IP address information.

OFF disables the oscilloscope to search the network for a DHCP server.

Examples

ETHERNET:DHCPBOOTP ON sets the oscilloscope to search for a DHCP server and assign a dynamic IP address to the oscilloscope.

ETHERnet:DNS:IPADDress

This command specifies the network Domain Name Server (DNS) IP address.

Group Ethernet

Syntax ETHERnet:DNS:IPADDress <QString>
ETHERnet:DNS:IPADDress?

Arguments <QString> is a standard IP address value, enclosed in quotes.

Examples ETHERNET:DNS:IPADDRESS "128.196.13.252" sets the DNS IP address that the oscilloscope uses to communicate with the network.

ETHERnet:DOMAINname

This command specifies the network domain name.

Group Ethernet

Syntax ETHERnet:DOMAINname <QString>
ETHERnet:DOMAINname?

Arguments <QString> is the network domain name, enclosed in quotes.

Examples ETHERNET:DOMAINNAME "Alpha1.Mycorp.com" sets the domain name that the oscilloscope uses to communicate with the network.
**ETHERnet:ENET:ADDress? (Query Only)**

Returns the Ethernet address (MAC address) value assigned to the oscilloscope. This is assigned at the factory and can not be changed.

**Group**  
Ethernet

**Syntax**  
ETHERnet:ENET:ADDress?

**Examples**  
ETHERNET:ENET:ADDRESS? returns an Ethernet address such as 08:00:11:01:02:03

**ETHERnet:GATEWay:IPADDress**

This command specifies the network gateway IP address.

**Group**  
Ethernet

**Syntax**  
ETHERnet:GATEway:IPADDress <QString>  
ETHERnet:GATEway:IPADDress?

**Related Commands**  
ETHERnet NETWORKCONFig  
ETHERnet IPADDress  
ETHERnet SUBNETMask

**Arguments**  
<QString> is a standard IP address value, enclosed in quotes.

**Examples**  
ETHERNET:GATEWAY:IPADDRESS "128.143.16.1" sets the gateway IP address.

**ETHERnet:IPADDress**

This command sets the IP address assigned to the oscilloscope.

**Group**  
Ethernet
Commands listed in alphabetical order

Syntax
ETHERnet:IPADDress <QString>
ETHERnet:IPADDress?

Related Commands
ETHERnet:NETWORKCONFig
ETHERnet:SUBNETMask
ETHERnet:GATEWay:IPADDress

Arguments
<QString> is a standard IP address value, enclosed in quotes.

Examples
ETHERNET:IPADDRESS "123.121.13.214" sets the oscilloscope's IP address.

ETHERnet:LXI:LAN:RESET (No Query Form)

This command resets the LXI local area network.

Group
Ethernet

Syntax
ETHERnet:LXI:LAN:RESET

Examples
ETHERnet:LXI:LAN:RESET resets the LXI local area network.

ETHERnet:LXI:LAN:SERVICENAME

This command sets or queries the service name used for the LXI interface.

Group
Ethernet

Syntax
ETHERnet:LXI:LAN:SERVICENAME <QString>
ETHERnet:LXI:LAN:SERVICENAME?

Arguments
<QString> is a quoted string of up to 64 characters that specifies the mDNS service name used for the LXI interface.

Examples
ETHERNET:LXI:LAN:SERVICENAME? might return "Tektronix Oscilloscope MSO5 053CVV"
ETHERnet:LXI:LAN:STATus? (Query Only)

This query returns the LXI network status: one of OK, FAULT, or IDENTIFY. IDENTIFY indicates that the device identify mode is enabled.

Group
Ethernet

Syntax
ETHERnet:LXI:LAN:STATus?

Related Commands
ETHERnet:PING
ETHERnet:PING:STATus?

Returns
OK — indicates the network is running and the oscilloscope can “see” the network.
FAULT — indicates the network is not visible, or the network settings are incorrect.
IDENTIFY indicates that a message is being displayed on the oscilloscope’s front panel indicating which scope the user is currently accessing.

Examples
ETHER:LXI:LAN:STAT? might return FAULT, indicating the network is not visible, or the network settings are incorrect.

ETHERnet:NAME

This command sets or queries the instrument Ethernet hostname assigned to the oscilloscope.

Group
Ethernet

Syntax
ETHERnet:NAME <QString>
ETHERnet:NAME?

Arguments
<QString> is the network name assigned to the oscilloscope, enclosed in quotes.

Examples
ETHERNET:NAME "labscope1" sets the oscilloscope's network name.

ETHERnet:NETWORKCONFig

This command specifies the Ethernet network configuration setting.
**ETHERnet:NETWORKCONFig**

**Syntax**

```plaintext
ETHERnet:NETWORKCONFig {AUTOMATIC|MANual}
ETHERnet:NETWORKCONFig?
```

**Related Commands**

- `ETHERnet:NAME`
- `ETHERnet:IPADDress`
- `ETHERnet:SUBNETMask`
- `ETHERnet:GATEWay:IPADDress`
- `ETHERnet:DHCPbootp`

**Arguments**

- **AUTOMATIC** specifies that the oscilloscope’s IP address, subnet mask and gateway settings will be received from a DHCP server on the local network.
- **MANual** specifies that the Ethernet settings will be configured manually, using `ETHERnet:IPADDress`, `ETHERnet:SUBNETMask`, and `ETHERnet:GATEWay:IPADDress`.

**Examples**

- `ETHERnet:NETWORKCONFig MANual` specifies to configure the Ethernet settings manually.
- `ETHERnet:NETWORKCONFig?` might return AUTOMATIC, indicating the settings are being configured automatically.

---

**ETHERnet:PING (No Query Form)**

Sends a ping packet to the instrument gateway and sets the status accordingly.

**Syntax**

```plaintext
ETHERnet:PING EXECute
```

**Examples**

- `ETHERNET:PING EXECute` causes the oscilloscope to ping the gateway IP address.

---

**ETHERnet:PING:STATUs? (Query Only)**

Returns the results of sending the `ETHERnet:PING` command to ping the gateway IP address.
Commands listed in alphabetical order

**ETHERnet:PING:STATus?**

Group: Ethernet

Syntax: `ETHERnet:PING:STATus?`

Returns:
- `OK` is returned if the computer at the gateway IP address answers.
- `NORESPONSE` is returned if the computer at the gateway IP address does not answer.
- `TRYING` is returned if the ping operation is still executing.
- `NEVER` is returned if `ETHERnet:PING EXECute` has not been previously sent, indicating that no ping response has ever been received in response to an `ETHERnet:PING EXECute` command since the instrument was powered on.

**ETHERnet:SUBNETMask**

This command sets or queries the instrument subnet mask value.

Group: Ethernet

Syntax: `ETHERnet:SUBNETMask <QString>`

Related Commands:
- `ETHERnet:NETWORKCONFig`
- `ETHERnet IPADDress`
- `ETHERnet GATEway:IPADDress`

Arguments:
- `<QString>` is the subnet mask value, enclosed in quotes.

Examples:
- `ETHERNET:SUBNETMASK "255.255.255.0"` sets the subnet mask value using standard IP address notation format.

**EVENT? (Query Only)**

This query-only command returns an event code from the Event Queue that provides information about the results of the last `*ESR?` read. `EVENT?` also removes the returned value from the Event Queue.

Group: Status and Error
Syntax

EVENT?

Related Commands

ALLEv?
*CLS
DESE
*ESE
*ESR?
EVMsg?
*SRE
*STB?

Examples

EVENT? might return :EVENT 110, showing that there was an error in a command header.

EVMsg? (Query Only)

This query-only command removes a single event code from the Event Queue that is associated with the results of the last *ESR? read and returns the event code with an explanatory message. For more information, see Event Handling.

Group

Status and Error

Syntax

EVMsg?

Related Commands

ALLEv?
*CLS
DESE
*ESE
*ESR?
EVENT?
*SRE
*STB?

Returns

The event code and message in the following format:
Commands listed in alphabetical order

where <Command> is the command that caused the error and may be returned when a command error is detected by the instrument. As much of the command will be returned as possible without exceeding the 60 character limit of the <Message> and <Command> string combined. The command string is right-justified.

Examples EVMSG? might return :EVMSG 110,"Command header error".

**EVQty? (Query Only)**

This query-only command returns the number of events that are enabled in the queue. This is useful when using the ALLEv? query, since it lets you know exactly how many events will be returned.

**Group** Status and Error

**Syntax** EVQty?

**Related Commands** ALLEv?  
EVENT?  
EVMsg?

Examples EVQTY? might return :EVQTY 3, indicating the number of event codes in the Event Queue.

**EYEMASK:MASK<x>:COUNt:HITS? (Query Only)**

This command returns the total number of hit violations for all segments in the specified mask test.

**Conditions** Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

**Group** Mask

**Syntax** EYEMASK:MASK<x>:COUNt:HITS?
Commands listed in alphabetical order

**Related Commands**

PLOT.PLOT<x>:MASK?

**Arguments**

MASK<x> is the number of the specified mask test in an eye diagram plot.

**Examples**

EYEMASK:MASK2:COUNT:HITS? might return 211, indicating that mask 2 has a total of 211 violations for all segments in the mask.

**EYEMASK:MASK<x>:COUNT:SEG<y>:HITS? (Query Only)**

This command returns the number of hit violations for the specified segment (area).

**Conditions**

Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

**Group**

Mask

**Syntax**

EYEMASK:MASK<x>:COUNT:SEG<y>:HITS?

**Related Commands**

PLOT.PLOT<x>:MASK?

**Arguments**

MASK<x> is the number of the specified mask test (or mask test plot?).

SEG<y> is the number of the mask segment for which to return hit violations data.

**Returns**

Returns an integer number representing the number of mask test violations for the specified mask test and mask segment.

**Examples**


**EYEMASK:MASK<x>:CREATor? (Query Only)**

This query-only command returns the name of the eye diagram plot that created the mask.

**Conditions**

Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).
Commands listed in alphabetical order

Group     Mask

Syntax     EYEMASK:MASK<x>:CREATor?

Related Commands     PLOT:PLOT<x>:MASK?

Arguments     MASK<x> is the number of the specified mask test (or mask test plot?).

Returns     <QString> is a quoted string that contains the name of the eye diagram plot associated with the specified mask.

Examples     EYEMASK:MASK4:CREATor? might return "plot2".

EYEMASK:MASK<x>:ENAbled

This command enables or disables eye mask testing in the specified plot.

Conditions     Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group     Mask?

Syntax     EYEMASK:MASK<x>:ENAbled {ON|OFF}
            EYEMASK:MASK<x>:ENAbled?

Related Commands     PLOT:PLOT<x>:MASK?

Arguments     MASK<x> is the number of the specified mask test (or mask test plot?).

Returns     Returns an integer number representing enabled (1) or disabled (0).

Examples     EYEMASK:MASK4:ENABLEd ON enables the eye mask test in the specified mask plot.
              EYEMASK:MASK2:ENABLEd? might return 0, indicating that mask testing is disabled in the specified mask plot.
**EYEMASK:MASK<x>:MASKfile**

This command sets or queries the current mask definition file name for the specified mask test.

**Conditions**
Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

**Group**
Mask

**Syntax**

```
EYEMASK:MASK<x>:MASKfile <Qstring>
EYEMASK:MASK<x>:MASKfile?
```

**Related Commands**

PLOT:PLOT<x>:MASK?

**Arguments**

- `<x>` is the number of the specified mask test (or mask test plot?).
- `<Qstring>` is a quoted string that defines the file path that specifies the location of the mask file to use, in the format `'[<path>]<filename.ext]'`. Specifying a path is optional. If no path is entered, the instrument will search in the current working directory as set in FILESystem CWD.

  - If the file argument begins with a file path separator (forward slash character) or a drive designator such as C:, then the file name is interpreted as a full path.
  
  - If the file argument begins with "." or "." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

**Examples**

```
EYEMASK:MASK1:MASKfile "'/C:/Applications/Masks/SerialRapidIO/RIO Serial 2_5G.msk" sets mask1 to use the RIO serial 2.5G mask file at the specified location

EYEMASK:MASK4:MASKfile? might return
"/media/C:/Applications/Masks/1394b/S400b_T1.msk"
```

**EYEMASK:MASK<x>:TEST:SAMPLE:THRESHOLD**

This command sets or queries the total number of hit violations that will cause a mask test failure.

**Conditions**
Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).
Commands listed in alphabetical order

**EYEMASK:**

<table>
<thead>
<tr>
<th>Group</th>
<th>Mask</th>
</tr>
</thead>
</table>

**Syntax**

EYEMASK:MASK<x>:TEST:SAMPLE:THRESHOLD <NR1>
EYEMASK:MASK<x>:TEST:SAMPLE:THRESHOLD?

**Related Commands**

PLOT:PLOT<x>:MASK?

**Arguments**

MASK<x> is the number of the specified mask test (or mask test plot?).

<NR1> is a positive integer indicating the number of mask hits required to cause a fail condition for that mask test.

**Examples**

EYEMASK:MASK1:TEST:SAMPLE:THRESHOLD 15 sets the specified mask test to fail if 15 or more mask violations occur.

EYEMASK:MASK3:TEST:SAMPLE:THRESHOLD? might return 20, indicating that the specified mask test is set to fail if 20 or more mask violations occur.

**EYEMASK:**

**MASK<x>:TEST:STATUS? (Query Only)**

This query-only command returns the mask hit test status.

**Conditions**

Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

**Group**

Mask

**Syntax**

EYEMASK:MASK<x>:TEST:STATUS?

**Related Commands**

PLOT:PLOT<x>:MASK?

**Arguments**

MASK<x> is the number of the specified mask test (or mask test plot?).

**Returns**

This command returns a string with the mask test status. Valid status strings are:

- **PASS.** The number of mask hits is less than the target fail threshold.
- **FAIL.** The number of mask hits is greater than or equal to the target fail threshold.
- **OFF.** Mask testing disabled on the specified mask plot.
Examples

EYEMASK:MASK2:TEST:STATUS? might return Fail, indicating that mask testing has failed in the specified mask plot.

FACTory (No Query Form)

This command (no query form) resets the instrument to its factory default settings. This command is equivalent to pressing the DEFAULT SETUP button located on the instrument front panel or selecting Default Setup from the File menu.

This command Performs the following in addition to what is done for the *RST command:

- Clears any pending OPC operations.
- Resets the following IEEE488.2 registers:
  - *ESE 0 (Event Status Enable Register)
  - *SRE 0 (Service Request Enable Register)
  - DESE 255 (Device Event Status Enable Register)
  - *PSC 1 (Power-on Status Clear Flag)
- Deletes all defined aliases.
- Enables command headers (:HEAder 1).

Group

Save and Recall

Syntax

FACTory

Related Commands

*PSC
RECALL:SETUp
*RST

Arguments

None

Examples

FACTORY resets the instrument to its factory default settings.

FILESystem? (Query Only)

This query-only command returns the directory listing of the current working directory. This query is the same as the FILESystem:DIR? query.
Group: File System

Syntax: FILESystem?

Related Commands:
- FILESystem:COPy
- FILESystem:CWD
- FILESystem:DELEte
- FILESystem:DIR?
- FILESystem:READFile
- FILESystem:REName
- FILESystem:WRITEFile

Arguments: None.

Examples: FILESYSTEM? might return :FILESYSTEM:DIR "myFile.txt","mywaveform.wfm".

**FILESystem:COPy (No Query Form)**

This command (no query form) copies a named file to a new file. The new file might be in a totally separate directory than the old file. You can only copy one file at a time using this command. Wild card characters are not allowed.

Group: File System

Syntax: FILESystem:COPy {<source_file_path>,<destination_file_path>}

Related Commands:
- FILESystem:CWD
- FILESystem:DELEte

Arguments:

- `<source_file_path>` is a quoted string that defines the file name and path or directory. If the file path is within the current working directory, you need only specify the file name.

- `<destination_file_path>` is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.
Examples

- `FILESYSTEM:COPY "E:/setup1.set","E:/SETUPS/setup1.set"` copies the file named setup1.set, on the E drive to a file named setup1.set in the SETUPS directory on the E drive.

- `:FILESystem:CWD "E:/SETUPS"`

- `:FILESystem:COPY ".","I:/Archive/SETUPS"`

- `:FILESystem:COPY "/ch1.isf","I:/SavedWfms/ch1_new.isf"`

**FILESystem:CWD**

This command sets or queries the current working directory. CWD is short for Current Working Directory. It changes the directory (folder) that the other FILESystem commands operate on.

**NOTE.** There are three host ports on the front panel (E:, F:, and G:), and two on the back panel (H: and I:).

**Group** File System

**Syntax**

```
FILESystem:CWD {<new_working_directory_path>}
FILESystem:CWD?
```

**Arguments**

- `<new_working_directory_path>` is a quoted string that defines the current working directory name. The maximum length of the directory name is 128 characters.

**Examples**

- `FILESYSTEM:CWD "E:/waveforms/"` changes the current working directory to a folder named "waveforms" on the USB flash drive installed in the "E:" USB Host port.

- `FILESYSTEM:CWD?` might return :FILESYSTEM:CWD "E:/" Indicates the current working directory is the root folder of the E: USB flash drive port.

**FILESystem:DELeete (No Query Form)**

This command (no query form) deletes a named file or directory from a mass storage device. Once removed, the data in that file or directory can no longer be accessed. If the specified file is a directory, it must be empty before it can be deleted.

**Group** File System
Commands listed in alphabetical order

**Syntax**

FILESystem:DELeTe <file_path>

**Related Commands**

FILESystem:COPy
FILESystem:CWD
FILESystem:RMDir

**Arguments**

<file_path> is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

**Examples**

FILESYSTEM:DELETE "NOT_MINE.SET" deletes the file named NOT_MINE.SET from the folder referred to by the FILESYSTEM:CWD command.

**FILESystem:DIR? (Query Only)**

This query-only command returns a comma separated list of quoted strings. Each string contains the name of a file or directory in the folder referred to by the FILESYSTEM:CWD command.

**Group**

File System

**Syntax**

FILESystem:DIR?

**Related Commands**

FILESystem:CWD
FILESystem:MKDir

**Arguments**

None

**Examples**

FILESYSTEM:DIR? might return :FILESYSTEM:DIR
"161012_132039_000.wfm","161012_132039_001.wfm","161220_191452.png",
"161220_191554.csv","170320_132925_000.set","170320_132929_000.set",
"Ch2-Ch3_000.set","E:","F:","G:","H:","I:","J:","Rgjtest_000.set",
"RgjSetupRuntTrig4chnl
s_000.set","RgjSetup_000.set","ScrnShotjeh2_000.wfm","ScrnShotljeh_000.wfm",
"ScrnShot_000.png","ScrnShot_000.wfm","Scrn
_000.set","Scrn_001.set","Scrn_002.set","Scrn_003.set","Scrn_004.set",
"Wfm_000.wfm","Wfm_001.wfm","Wfm_002.wfm","Wfm_003
.wfm","Wfm_004.wfm","Wfm_005.wfm","Wfm_006.wfm","digChans_000.wfm".

2-380 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
FILESystem:HOMEDir? (Query Only)

This query returns the current user's home directory.

Group: File System

Syntax: FILESystem:HOMEDir?

Returns: The current user's home directory as a quoted string.

Examples:
FILESYSTEM:HOMEDIR? might return: "C:".

FILESystem:LDIR? (Query Only)

Returns a comma separated list of every file, file size, type, modification date and time, and directory in the folder referred to by the FILESYSTEM:CWD command. This is different than the :DIR query in that it provides a long output format with the file size, type, and modification date/time. Each entry is a semicolon separated list: <filename>;<type>;<size in bytes>;<date>;<time>

Group: File System

Syntax: FILESystem:LDIR?

Returns: A comma separated list of every file, file size, type, modification date and time, and directory in the folder referred to by the FILESYSTEM:CWD command.

Examples:
FILESYSTEM:LDIR? might return "tek0000CH1.isf;FILE;20342;2009-05-21;13:58:24", "TEMP;DIR;4096;2009-09-15;06:20:44".

FILESystem:MKDir (No Query Form)

This command (no query form) creates a new directory.

Group: File System

Syntax: FILESystem:MKDir <directory_path>
Related Commands

- FILESystem:CWD
- FILESystem:DIR?

Arguments

- `<directory_path>` is a quoted string that specifies the directory to create.

Examples

FILESYSTEM:MKDIR "E:\NewDirectory" creates the directory named `NewDirectory` at the root of the E drive.

These two commands create the directory `MyNewSubDirectory` within the existing directory `MyDirectory` at the root of the E drive:


FILESystem:READFile (No Query Form)

This command writes the contents of the specified file to the current interface. If the specified file does not exist or is not readable, an appropriate error event is posted.

Group

File System

Syntax

FILESystem:READFile <QString>

Related Commands

- FILESystem:CWD

Arguments

- `<QString>` is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

Examples

FILESYSTEM:READFILE "E:/test_data/tek00016CH1.csv" reads the content of the specified file, if the file exists and is readable, and sends the content of the file to the current interface.

FILESystem:REName (No Query Form)

This command (no query form) assigns a new name to an existing file or folder.

Group

File System
Commands listed in alphabetical order

Syntax  
FILESystem:REName <old_file_path>,<new_file_path>

Related Commands  
FILESystem CWD

Arguments  
<old_file_path> is a quoted string that defines the file or folder name and path. If the path is within the current working directory, you need only specify the file or folder name.

<new_file_path> is a quoted string that defines the file or folder name and path. If the path is within the current working directory, you need only specify the file or folder name.

Examples  
FILESYSTEM:RENAME "E:/TEK00000.SET","E:/MYSETTING.SET" gives the file named TEK00000.SET the new name of MYSETTING.SET. The file remains in the root directory on the E drive.

FILESYSTEM:RENAME "e:/mySettings/tek00000.set","e:/setup1.set". This example illustrates how to move a file from one folder to another.

FILESystem:RMDir (No Query Form)  
This command (no query form) deletes a named directory. The directory must be empty.

Group  
File System

Syntax  
FILESystem:RMDir <directory_path>

Related Commands  
FILESystem CWD

Arguments  
<directory_path> is a quoted string that defines the folder name and path. If the folder path is within the current working directory, you need only specify the folder name.

Examples  
FILESYSTEM:RMDIR "E:/oldDirectory" removes the directory named OldDirectory from the root of the E drive.
FILESystem:UNMOUNT:DRIve (No Query Form)

This command unmounts the USB drive specified by the quoted string argument.

Group File System

Syntax FILESystem:UNMOUNT:DRIve <QString>

Related Commands

Arguments <QString> is a quoted string that specifies which USB drive to unmount. String is a case insensitive single letter followed by a colon.

Examples FILESYSTEM:UNMOUNT:DRIVE "G:" specifies to unmount the flash drive installed in the right-most USB slot on the front of the instrument.

FILESystem:WRITEFile (No Query Form)

This command (no query form) writes the specified block data to the specified file on the instrument's file system. If the destination file cannot be written, an error event is posted.

Group File System

Syntax FILESystem:WRITEFile <file_path>,<data>

Related Commands FILESystem:CWD

Arguments <file_path> is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name. <data> is the specified block data to be written.

FPAne:PRESS (No Query Form)

This command is used to emulate a button press. When used with knob enumerations, this command pushes the knob. Use the FPAne:TURN command to emulate knob turns.
Commands listed in alphabetical order

**Group**  Miscellaneous

**Syntax**

```
FPAnel:PRESS {AUTOset|BUS|CH1<x>|CLEAR|CURsor|DEFaultsetup|FASTAcq|FORCEtrig|GPKNOB1|GPKNOB2|HIGHRES|HORZPOS|HORZScale|MATH|NEXT|PREV|REF|RUNStop|SETTO50|SINGleseq|TOUCHSCReen|TRIGMode|TRIGSlope|USER|VERTPOS|VERTSCALE|ZOOM}
```

**Arguments**  Arguments are instrument buttons.

**Examples**

```
FPANEL:PRESS FORCETRIG emulates pressing the Force trigger button.
```

**FPAnel:TURN**

This command is used to emulate a knob turn. The optional NR1 specifies the number of clicks where negative values indicate counter clockwise. If not specified, the default of 1 click is used indicating the knob is turned clockwise 1 click.

**Group**  Miscellaneous

**Syntax**

```
FPAnel:TURN {GPKNOB1|GPKNOB2|HORZPOS|HORZScale|PANKNOB|TRIGLevel|VERTPOS|VERTSCALE|ZOOM} [,<NR1>]
```

**Arguments**  Arguments are knobs that turn obtained during startup.

<NR1> is the number of clicks to turn the knob.

**Examples**

```
FPANEL:TURN TRIGLEVEL ,3 emulates turning the trigger Level knob 3 clicks in the clockwise direction.
```

**HEADer**

This command sets or queries the Response Header Enable State that causes the instrument to either include or omit headers on query responses.

**NOTE.** This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk); these commands never return headers.

Whether the long or short form of header keywords and enumerations are returned is dependent upon the state of .VERBose.
Commands listed in alphabetical order

**Group**  
Miscellaneous

**Syntax**  
HEADer {<NR1>|OFF|ON}  
HEADer?

**Related Commands**  
VERBose

**Arguments**  
<NR1> = 0 sets the Response Header Enable State to false; any other value sets this state to true.

OFF sets the Response Header Enable State to false. This causes the instrument to omit headers on query responses, so that only the argument is returned.

ON sets the Response Header Enable State to true. This causes the instrument to include headers on applicable query responses. You can then use the query response as a command.

**Examples**  
HEADER OFF specifies that the instrument omits headers on query responses, so that only the argument is returned.

HEADER? might return :HEADER 1, indicating that the instrument is including headers on applicable query responses.

With :VERBose ON and :HEADER ON, the :ACQuire:MODe? query might return :ACQUIRE:MODE SAMPLE.

With :VERBose OFF and :HEADER ON, the :ACQuire:MODe? query might return :ACQ:MOD SAM

With :VERBose ON and :HEADER OFF, the :ACQuire:MODe? query might return SAMPLE

With :VERBose OFF and :HEADER OFF, the :ACQuire:MODe? query might return SAM

**HORizontal? (Query Only)**  
Queries the current horizontal settings.

**Group**  
Horizontal

**Syntax**  
HORizontal?
HORizontal:ACQDURATION? (Query Only)

This query returns the timebase duration.

Group  Horizontal

Syntax  HORizontal:ACQDURATION?

Returns  <NR3> returns the duration of the acquisition.

Examples  HORizontal:ACQDURATION? might return :HORIZONTAL:ACQDURATION 5.0E-9, indicating the acquisition duration is 5.0 us.

HORizontal:DELay:MODE

This command sets or queries the horizontal delay mode.

Group  Horizontal

Syntax  HORizontal:DELay:MODE {OFF|ON|<NR1>}

HORizontal:DELay:MODE?

Related Commands  HORizontal POSition

Arguments  OFF sets the Horizontal Delay Mode to off. This causes the HORizontal:POSition command to operate like the HORIZONTAL POSITION knob on the front panel.

ON sets the Horizontal Delay Mode to on. This causes the HORizontal:DELay:TIME command to operate like the HORIZONTAL POSITION knob on the front panel.
Commands listed in alphabetical order

<NR1> = 0 sets the Horizontal Delay Mode to off, any other value sets this mode to on.

Examples

HORizontal:DELAY:MODE OFF sets the Horizontal Delay Mode to off, allowing the horizontal position command to operate like the HORIZONTAL POSITION knob on the front panel.

HORizontal:DELAY:MODE? might return HORizontal:DELAY:MODE OFF indicating that the Horizontal Delay Mode is off and that the horizontal position command operates like the HORIZONTAL POSITION knob on the front panel.

HORizontal:DELay:TIme

This command sets or queries the horizontal delay time that is used when delay mode is on.

Group

Horizontal

Syntax

HORizontal:DELay:TIme <NR3>
HORizontal:DELay:TIme?

Related Commands

HORizontal:POSition

Arguments

NR3 is the delay in seconds.

Examples

HORizontal:DELay:TIme 0.3 sets the delay of acquisition data so that the resulting waveform is centered 300 ms after the trigger occurs.

HORizontal:DIVisions? (Query Only)

This query-only command returns the number of graticule divisions.

Group

Horizontal

Syntax

HORizontal:DIVisions?

Examples

HORizontal:DIVisions? might return :HORIZONTAL:DIVISIONS 10.0000, indicating that the waveform is displayed across ten divisions.
**HORizontal:FASTframe? (Query Only)**

This query returns all information under horizontal:fastframe.

**Group**  
Horizontal

**Syntax**  
HORizontal:FASTframe?

**Examples**  
HORIZONTAL:FASTFRAME? might return  
:HORIZONTAL:FASTFRAME:SELECTED 1;SUMFRAME:STATE 0;HORIZONTAL:FASTFRAME:COUNT 2;MULTIPLEFRAMES:MODE OFF;:HORIZONTAL:FASTFRAME:STATE 0;REF:INCLUDE 0;FRAME 1, the current FastFrame settings.

**HORizontal:FASTframe:COUNt**

This command sets or returns the number of frames.

**Group**  
Horizontal

**Syntax**  
HORizontal:FASTframe:COUNt <NR1>  
HORizontal:FASTframe:COUNt?

**Arguments**  
<NR1> is the number of frames.

**Examples**  
:HORIZONTAL:FASTframe:COUNt 10 sets the number of frames to be acquired to 10. If this is beyond the maximum number of frames, the value will be constrained. For example, if 8 is the current maximum (based on record length) 8 will be set instead of 10.

**HORizontal:FASTframe:MAXFRames? (Query Only)**

This query returns the maximum number of frames.

**Group**  
Horizontal

**Syntax**  
HORizontal:FASTframe:MAXFRames?
Examples  
HORIZONTAL:FASTFRAME:MAXFRAMES? might return  
:HORIZONTAL:FASTFRAME:MAXFRAMES 2, indicating the maximum number  
of frames is 2.

HORIZONTAL:FASTframe:MULtipleframes:MODE

This command sets or returns the overlay display type.

Group  
Horizontal

Syntax  
HORIZONTAL:FASTframe:MULtipleframes:MODE {OFF|OVERlay}  
HORIZONTAL:FASTframe:MULtipleframes:MODE?

Arguments  
OFF specifies only displaying the selected frame.  
OVERlay specifies overlaying all frames with the temperature palette. The  
summary frame is not included in the overlay. The selected frame is drawn in  
blue on top of all other frames.

Examples  
:HORIZONTAL:FASTframe:MULtipleframes:MODE OVERlay will cause the  
entire acquired set of frames to be rendered in a single image, similar to how  
FastAcq appears. The currently selected frame is drawn on top in blue.

HORIZONTAL:FASTframe:REF:FRAME

This command sets or returns the reference frame number.

Group  
Horizontal

Syntax  
HORIZONTAL:FASTframe:REF:FRAME <NR1>  
HORIZONTAL:FASTframe:REF:FRAME?

Arguments  
<NR1> is the reference frame number.

Examples  
HORIZONTAL:FASTFRAME:REF:FRAME? might return  
:HORIZONTAL:FASTFRAME:REF:FRAME 1, indicating the reference frame  
number is 1.
HORizontal:FASTframe:REF:INCLUDe

This command sets or returns whether the reference frame delta information is shown in the display.

Group  Horizontal

Syntax  
HORizontal:FASTframe:REF:INCLUDe {<NR1>|OFF|ON}
HORizontal:FASTframe:REF:INCLUDe?

Arguments  ON displays the delta information.
OFF does not display the delta information.
<NR1> a 0 indicates the delta information is off, any other value displays the delta information.

HORizontal:FASTframe:SELECTED

This command sets or returns the selected frame number for acquired frames. Refs have their own selected frames.

Group  Horizontal

Syntax  
HORizontal:FASTframe:SELECTED <NR1>
HORizontal:FASTframe:SELECTED?

Arguments  <NR1> is the selected frame number for acquired frames.

Examples  
HORIZONTAL:FASTFRAME:SELECTED? might return 
:HORIZONTAL:FASTFRAME:SELECTED 1, indicating the selected frame number is 1.

HORizontal:FASTframe:STATE

This command sets or returns the state of FastFrame. Acquisition modes Envelope and Average are not compatible with FastFrame. If FastFrame is on, an attempted set to those acquisition modes will fail and revert to Sample mode. If FastFrame is turned on while in one of those acquisition modes, the acquisition mode is changed to Sample.
Commands listed in alphabetical order

Group Horizontal

Syntax

HORizontal:FASTframe:STATE {<NR1>|OFF|ON}
HORizontal:FASTframe:STATE?

Arguments

ON indicates FastFrame is active.
OFF indicates that FastFrame is off.
<NR1> A 0 turns off FastFrame, any other value activates FastFrame.

Examples

HORIZONTAL:FASTFRAME:STATE? might return
:HORIZONTAL:FASTFRAME:STATE 0, indicating FastFrame is off.

HORizontal:FASTframe:SUMFrame? (Query Only)

This command sets or returns the summary frame type. Turning on Summary Frame does not adjust the numberFrames value as long as there is room for an additional frame. If there is not enough room then numberFrames will be reduced by 1. The numberFrames value is always the number of frames to acquire.

Group Horizontal

Syntax

HORizontal:FASTframe:SUMFrame? {NONE|AVERAGE|ENVELOPE}
HORizontal:FASTframe:SUMFrame??

Arguments

NONE sets the Summary frame to off.
AVERAGE sets the Summary frame to average of all acquired frames.
ENVELOPE sets the Summary frame to envelope of all acquired frames.

Examples

HORIZONTAL:FASTFRAME:SUMFRAME? might return
:HORIZONTAL:FASTFRAME:SUMFRAME NONE, indicating the summary frame is off.

HORizontal:FASTframe:SUMFrame:STATE

This command sets or returns the state of FastFrame summary frame. Summary frame mode is set automatically based on the acquisition mode. When in Sample mode, the summary frame type is set to Average. When in Peak Detect mode, the
summary frame type is set to Envelope. When in High Res mode, the summary frame type is set to Average.

**Group** Horizontal

**Syntax**

```
HORIZONTAL:FASTframe:SUMFrame:STATE {<NR1>|OFF|ON}
HORIZONTAL:FASTframe:SUMFrame:STATE?
```

**Arguments**

- **ON** indicates summary frame is active.
- **OFF** indicates that summary frame is off.
- `<NR1>` a 0 turns off summary frame; any other value activates the summary frame.

**Examples**

HORIZONTAL:FASTFRAME:SUMFRAME:STATE? might return:

```
:HORIZONTAL:FASTFRAME:SUMFRAME:STATE 0, indicating the summary frame is off.
```

**HORizontal:FASTframe:TIMEStamp:ALL? (Query Only)**

This query-only command returns the time stamp of all frames. The format is (Frame #: TimeStamp, Frame #: TimeStamp, and so on). Each time-stamp string is of the form `DD.MM.YYYY.HH:MM::SS.xxxxxxxxxxxx`.

**Group** Horizontal

**Syntax**

```
HORIZONTAL:FASTframe:TIMEStamp:ALL?
```

**Returns** Returns all of the timestamps.

**Examples**

HORIZONTAL:FASTFRAME:TIMESTAMP:ALL? might return:

```
:HORIZONTAL:FASTFRAME:TIMESTAMP:ALL "1:31.12.1969.16:00:00.000000000000", indicating the only time
```

**HORizontal:FASTframe:TIMEStamp:DELTa? (Query Only)**

This query returns the time difference between the Selected and Reference time-stamps.

**Group** Horizontal
Commands listed in alphabetical order

Syntax

HORIZONTAL:FASTframe:TIMEStamp:DELTa?

Returns

Return value is in seconds and fraction of a second.

Examples

HORIZONTAL:FASTFRAME:TIMESTAMP:DELTA? might return
:HORIZONTAL:FASTFRAME:TIMESTAMP:DELTA "0.000000000000"
indicating the time difference is 0.0 seconds.

HORIZONTAL:FASTframe:TIMEStamp:REReference? (Query Only)

This query returns the time-stamp of the FastFrame Reference frame.

Group Horizontal

Syntax

HORIZONTAL:FASTframe:TIMEStamp:REFERENCE?

Examples

HORIZONTAL:FASTFRAME:TIMESTAMP:REFERENCE? might return
:HORIZONTAL:FASTFRAME:TIMESTAMP:REFERENCE "31.12.1969.17:00:00.000000000000"
indicating the reference timestamp is 31.12.1969.17:00:00.000000000000.

HORIZONTAL:FASTframe:TIMEStamp:SELECTED? (Query Only)

This query returns the time-stamp of the FastFrame Selected acquired frame.

Group Horizontal

Syntax

HORIZONTAL:FASTframe:TIMEStamp:SELECTED?

Examples

HORIZONTAL:FASTFRAME:TIMESTAMP:SELECTED? might return
:HORIZONTAL:FASTFRAME:TIMESTAMP:SELECTED "31.12.1969.17:00:00.000000000000"
indicating the timestamp of the selected frame is 31.12.1969.17:00:00.000000000000.

HORIZONTAL:FASTframe:XZErO:ALL? (Query Only)

This query-only command returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the
waveform specified by the DATa SOurce command for all frames. This value is in units of WFMOutpre:XUnit. The format is a string of the form (frame #:xzero, frame #:xzero, and so on).

Group  | Horizontal

Syntax  | HORizontal:FASTframe:XZeRO:ALL?

Returns  | This returns the XZERO values for all of the frames.


**HORizontal:FASTframe:XZeRO:REF? (Query Only)**

This query-only command returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa SOurce command for the reference frame. This value is in units of WFMOutpre:XUnit.

Group  | Horizontal

Syntax  | HORizontal:FASTframe:XZeRO:REF?

Examples  | HORizontal:FASTFRAME:XZERO:REF? might return :HORIZONTAL:FASTFRAME:XZERO:REF 1.0 indicating the sub-sample time is 1.0.

**HORizontal:FASTframe:XZeRO:SELECTED? (Query Only)**

This query-only command returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa SOurce command for the selected frame. This value is in units of WFMOutpre:XUnit.

Group  | Horizontal
Commands listed in alphabetical order

**HORizontal:FASTframe:XZEro:SELECTED?**

Syntax

```
HORIZONTAL:FASTFRAME:XZERO:SELECTED?
```

Examples

```
HORIZONTAL:FASTFRAME:XZERO:SELECTED? might return
:HORIZONTAL:FASTFRAME:XZERO:SELECTED 1.0 indicating the sub-sample
time is 1.0.
```

**HORizontal:MAIn:INTERPRatio? (Query Only)**

This query-only command returns the Horizontal interpolation ratio.

Group

Horizontal

Syntax

```
HORIZONTAL:MAIN:INTERPRatio?
```

Examples

```
HORIZONTAL:MAIN:INTERPRATIO? might return
```

**HORizontal:MODE**

This command set or queries the horizontal operating mode.

Group

Horizontal

Syntax

```
HORizontal:MODE {AUTO|MANual}
HORizontal:MODE?
```

Related Commands

- HORizontal:MODE:RECorDlength
- HORizontal:MODE:SAmpLerate
- HORizontal:MODE:SCALE

Arguments

- AUTO selects the automatic horizontal model. Auto mode automatically adjusts the sample rate and record length to provide a high acquisition rate in Fast Acq or signal fidelity in analysis. Record length is read only.

- MANUAL selects the manual horizontal model. Manual mode lets you change the sample rate, horizontal scale, and record length. These values interact. For example, when you change record length then the horizontal scale also changes.
Examples  
HORIZONTAL:MODE AUTO sets the horizontal mode to auto.
HORIZONTAL:MODE? might return :HORIZONTAL:MODE MANUAL, indicating that the horizontal mode is manual.

**HORizontal:MODe:AUTOmatic:FASTAcq:RECORDlength:MAXimum:VALue**

Sets or queries the horizontal FastAcq maximum record length.

**Group**  
Horizontal

**Syntax**  
HORizontal:MODe:AUTOmatic:FASTAcq:RECORDlength:MAXimum:VALue <NR1>
HORizontal:MODe:AUTOmatic:FASTAcq:RECORDlength:MAXimum:VALue?

**Arguments**  
<NR1> is the horizontal FastAcq maximum record length.

**Examples**  

**HORizontal:MODe:AUTOmatic:FASTAcq:RECORDlength:MAXimum:ZOOMOVERride**

Sets or queries the flag which allows override of the horizontal FastAcq maximum record length.

**Group**  
Horizontal

**Syntax**  
HORizontal:MODe:AUTOmatic:FASTAcq:RECORDlength:MAXimum:ZOOMOVERride {OFF|ON|0|1}
HORizontal:MODe:AUTOmatic:FASTAcq:RECORDlength:MAXimum:ZOOMOVERride?

**Arguments**  
OFF does not allow override of the horizontal FastAcq maximum record length.
ON allows override of the horizontal FastAcq maximum record length.
0 does not allow override of the horizontal FastAcq maximum record length.
1 allows override of the horizontal FastAcq maximum record length.
Examples

HORIZONTAL:MODE:AUTOMATIC:FASTACQ:RECORDLENGTH:MAXIMUM:
ZOOMOVERRIDE 0 does not allow override of the horizontal FastAcq maximum
record length.

HORIZONTAL:MODE:AUTOMATIC:FASTACQ:
RECORDLENGTH:MAXIMUM:ZOOMOVERRIDE? might return
:HORIZONTAL:MODE:AUTOMATIC:FASTACQ:RECORDLENGTH:
MAXIMUM:ZOOMOVERRIDE 1 indicating that override of the horizontal FastAcq
maximum record length is allowed.

HORIZontal:MODe:MANual:CONFIGure

Sets or queries which horizontal control (scale or record length) will primarily
change when the sample rate is changed in Manual mode. If the selected control
(scale or record length) reaches a limit then the unselected control (record length
or scale) may also change.

Group      Horizontal

Syntax      HORIZontal:MODe:MANual:CONFIGure
            {HORIZontalscale|RECORDLength}
            HORIZontal:MODe:MANual:CONFIGure?

Arguments    HORIZontalscale will change when sample rate is adjusted.
              RECORDLength will change when sample rate is adjusted.

Examples     HORIZontal:MODe:MANual:CONFIGure RECORDLength allows the record
              length be adjusted when sample rate is changed in Manual mode.

HORIZontal:MODe:RECORDlength

This command sets or queries the record length.

Group      Horizontal

Syntax      HORIZontal:MODe:RECORDlength <NR1>
            HORIZontal:MODe:RECORDlength?

Arguments    <NR1> is the record length in samples. Manual mode lets you change the record
              length, while the record length is read only for Automatic mode.
Examples

HORIZONTAL:MODE:RECORDLENGTH 1000 sets the record length to 1000 samples.

HORIZONTAL:MODE:RECORDLENGTH? might return :HORIZONTAL:MODE:RECORDLENGTH 1000000, indicating that the record length is set to 1,000,000 samples.

HORIZONTAL:MODE:SAMPLERate

This command sets or queries the sample rate.

Group Horizontal

Syntax

HORIZONTAL:MODE:SAMPLERate <NR1>
HORIZONTAL:MODE:SAMPLERate?

Arguments

<NR1> is the sample rate in samples per second.

Examples

HORIZONTAL:MODE:SAMPLERATE 1e6 sets the sample rate to 1 million samples per second.

HORIZONTAL:MODE:SAMPLERATE? might return :HORIZONTAL:MODE:SAMPLERATE 5.0000E+6, indicating that the sample rate is set to 5 million samples per second.

HORIZONTAL:MODE:SCALE

This command sets or queries the horizontal scale.

Group Horizontal

Syntax

HORIZONTAL:MODE:SCALE <NR1>
HORIZONTAL:MODE:SCALE?

Arguments

<NR1> is the horizontal scale in seconds per division.

Examples

HORIZONTAL:MODE:SCALE 2e-9 sets the horizontal scale to 2 ns per division.

HORIZONTAL:MODE:SCALE? might return :HORIZONTAL:MODE:SCALE 20.0000E-6, indicating that the horizontal scale is set to 10 μs per division.
**HORizontal:POSition**

This command sets or queries the horizontal position as a percent of screen width. When Horizontal Delay Mode is turned off, this command is equivalent to adjusting the HORIZONTAL POSITION knob on the front panel. When Horizontal Delay Mode is turned on, the horizontal position is forced to 50%.

**Group**: Horizontal

**Syntax**

HORIZONTAL:POSITION <NR3>
HORIZONTAL:POSITION?

**Arguments**

<NR3> is from 0 to \( \approx 100 \) and is the position of the trigger point on the screen (0 = left edge, 100 = right edge).

**Examples**

HORIZONTAL:POSITION 10 sets the trigger position of the waveform such that 10% of the display is to the left of the trigger position.

HORIZONTAL:POSITION? might return :HORIZONTAL:POSITION 5.0000E+01 indicating the trigger point displayed on the screen.

**HORizontal:PREViewstate? (Query Only)**

This query returns the display system preview state.

**Group**: Horizontal

**Syntax**

HORIZONTAL:PREViewstate?

**Returns**

<NR1> = 1 if the system is in the preview state.

<NR1> = 0 if the system is not in the preview state.

**Examples**

HORIZONTAL:PREVIEWSTATE? might return :HORIZONTAL:PREVIEWSTATE 0 indicating the system is not in the preview state.

**HORizontal:RECOrdlength**

This command sets or queries the horizontal record length. To change the record length the Horizontal Mode must be set to Manual.
Commands listed in alphabetical order

**Horizontal**

**Group**

**Syntax**

HORIZONTAL:RECORDlength <NR1>
HORIZONTAL:RECORDlength?

**Arguments**

<NR1> is the horizontal record length.

**Examples**

HORIZONTAL:RECORDLENGTH 1000 sets the record length to 1000 samples.
HORIZONTAL:RECORDLENGTH? might return :HORIZONTAL:RECORDLENGTH 1000000, indicating that the record length is set to 1,000,000 samples.

**HORizontal:ROLL? (Query Only)**

Queries the horizontal roll mode status.

**Group**

Horizontal

**Syntax**

HORIZONTAL:ROLL?

**Returns**

ON indicates roll mode is active.
OFF indicates that summary frame is off.
<NR1> a 0 indicates roll mode is off, any other value activates roll mode.

**Examples**

HORIZONTAL:ROLL? might return :HORIZONTAL:ROLL 0 indicating roll mode is off.

**HORizontal:SAMPLERate**

This command sets or queries the horizontal sample rate.

**Group**

Horizontal

**Syntax**

HORIZONTAL:SAMPLERate <NR3>
HORIZONTAL:SAMPLERate?

**Arguments**

<NR3> is the horizontal sample rate in samples per second.
Examples

HORIZONTAL:SAMPLERATE 1e6 sets the sample rate to 1 million samples per second.

HORIZONTAL:SAMPLERATE? might return :HORIZONTAL:SAMPLERATE 5.0000E+6, indicating that the sample rate is set to 5 million samples per second.

HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRide

Sets or queries the flag which allows override of the horizontal analyze minimum sample rate.

Group

Horizontal

Syntax

HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRide
{OFF|ON|0|1}

HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRide?

Arguments

0 does not allow override of the horizontal analyze minimum sample rate.
1 allows override of the horizontal analyze minimum sample rate.
OFF does not allow override of the horizontal analyze minimum sample rate.
ON allows override of the horizontal analyze minimum sample rate.

Examples

HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:OVERRIDE OFF does not allow override of the horizontal analyze minimum sample rate.


HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue

Sets or queries the minimum sample rate used by Analysis Automatic horizontal mode.

Group

Horizontal

Syntax

HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue
{AUTOmatic|<NR3>}

HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue?
Related Commands

Arguments

AUTOmatic allows the oscilloscope to set the minimum value.

<NR3> is the minimum sample rate.

Examples

HORIZONTAL:SAMPLERate:ANALYZemode:MINimum:VALUE AUTOMATIC allows the oscilloscope to set the minimum value.

HORIZONTAL:SAMPLERate:ANALYZemode:MINimum:VALUE? might return
HORIZONTAL:SAMPLERate:ANALYZemode:MINimum:VALUE AUTOMATIC indicating the oscilloscope automatically sets the minimum value.

HORizontal:SCAle

This command sets or queries the horizontal scale.

Group
Horizontal

Syntax
HORIZONTAL:SCALE <NR3>
HORIZONTAL:SCALE?

Arguments

<NR3> is the horizontal scale in time per division.

Returns
The current horizontal scale is returned.

Examples
HORIZONTAL:SCALE 20e-9 sets the horizontal scale to 20 ns/division.

HORIZONTAL:SCALE? might return :HORIZONTAL:SCALE 20e-9 indicating the horizontal scale is set to 20 ns/division.

ID? (Query Only)

This query-only command returns identifying information about the instrument and related firmware similar to that returned by the *IDN? IEEE488.2 common query but does not include the instrument serial number.

Group
Miscellaneous

Syntax
ID?
Related Commands *IDN?

Examples *IDN? might return ID TEK/MSO54, CF:91.1CT, FV:1.2.0.2886, indicating that the instrument model number is set to MSO54, codes and formats is CF:91.1CT, and firmware version is FV:1.2.0.

*IDN? (Query Only)

This query-only command returns the instrument identification code.

Group Miscellaneous

Syntax *IDN?

Related Commands ID?

Examples *IDN? might return TEKTRONIX, MSO54, C100123, CF:91.1CT FV:1.2.0.2886, indicating the instrument model number, serial number, codes and formats number, and firmware version number.

LIC:UNINSTALL? (Query Only)

Returns the exit license indicated for the user to return to their TekAMS account. Active licenses can be specified by their nomenclature. TransactionIDs can be used to specify an active license or a previously uninstalled license. In either case, the exit-license is returned as block-data.

Group Miscellaneous

Syntax LIC:UNINSTALL? <QString>

Arguments <QString> is the nomenclature of an active license or a TransactionIDs to specify an active license or a previously uninstalled license.

Returns The exit-license is returned as block-data.

Examples LIC:UNINSTALL? “LIC5-SRAERO” uninstalls the given license and returns the license block data.
LICENSE? (Query Only)

This query-only command returns all license parameters.

Group  Miscellaneous
Syntax  LICENSE?


LICENSE:APPID? (Query Only)

This query returns a comma-separated list of the active application IDs. If a string argument is provided, a “0” or “1” is returned, according to whether the string matches an active application ID.

Group  Miscellaneous
Syntax  LICENSE:APPID? {<QString>}

Returns  This query returns a comma-separated list of the active application IDs. If a string argument is provided, a “0” or “1” is returned, according to whether the string matches an active application ID.

Examples  LIC:APPID? might return “BW5-2000, AFG, DVM, DJA”, which is a complete list of the active applications.
           LIC:APPID? “AFG” would return “1” because the app is active.

LICENSE:COUNT? (Query Only)

This query returns a count of the number of active licenses installed.

Group  Miscellaneous

LIC:UNINSTALL? “569765772” uninstalls the license with the given transaction ID and returns the license block data.
Syntax LICense:COUNT?

Returns A count of the number of active licenses installed.

Examples LICense:COUNT? might return :LICense:COUNT 2 indicating that 2 active licenses are installed

LICense:ERRor? (Query Only)

This query-only command prompts the instrument to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue. This command is an alias for ALLEv?.

Group Miscellaneous

Syntax LICense:ERRor?

Related Commands ALLEv?
*ESR?
EVMsg?

Examples LICense:ERRor? might return 0, "No events to report- queue empty".

LICense:GMT? (Query Only)

This query returns the GMT time in ISO 8601 format, the local date, 24 hour time and time-zone offset.

Group Miscellaneous

Syntax LICense:GMT?

Returns The GMT time in ISO 8601 format, the local date, 24 hour time and time-zone offset.
Examples  LICENSE:GMT? might return :LICENSE:GMT "2016-05-23T17:05:10-07:00" indicating the local GMT time.

**LICENSE:HID? (Query Only)**

This query returns the instrument HostID unique identifier.

Group    Miscellaneous

Syntax   LICENSE:HID?

Returns  The instrument HostID unique identifier.

Examples LICENSE:HID? might return :LICENSE:HID "TMS-9CS4US5SGJN6X"

**LICENSE:INSTALL (No Query Form)**

This command accepts a <block_data> license and installs it on the instrument. Restarting the instrument may be necessary to fully activate the additional capabilities.

Group    Miscellaneous

Syntax   LICENSE:INSTALL <block_data>

Arguments <block_data> is the license in block data format.

Examples LICENSE:INSTALL <block_data>

**LICENSE:ITEM? (Query Only)**

This query returns the details pertaining to a specific license. The NR1 argument is zero-indexed. If no argument is provided, zero is assumed.

Group    Miscellaneous
Commands listed in alphabetical order

Syntax

LICense:ITEM? <NR1>

Arguments

<NR1> is the zero-indexed argument specifying a specific license.

Returns

The details pertaining to a specific license.

Examples

LICense:ITEM? 0 might return
"5-BW-1000,Fixed,2116-06-15T14:55:54-07:00,11870047,BW5-1000,1 GHz bandwidth on 5 series oscilloscopes"

LICense:LIST? (Query Only)

This query returns the active license nomenclatures as a comma-separated list of strings. Duplicate nomenclatures, that is, the same license but with different expiration dates, are included.

Group

Miscellaneous

Syntax

LICense:LIST?

Returns

The active license nomenclatures as a comma-separated list of strings.

Examples

LICense:LIST? might return :LICENSE:LIST "5-BW-1000","SUP5-RL125M".

LICense:VALidate? (Query Only)

This query accepts a license nomenclature as an argument and returns True (1) if that nomenclature is active and any required hardware is installed, or False (0) if either the nomenclature is not active or required hardware is not installed.

Group

Miscellaneous

Syntax

LICense:VALidate? <QString>

Arguments

<QString> is the license nomenclature.
Returns True (1) if that nomenclature is active and any required hardware is installed. False (0) if either the nomenclature is not active or required hardware is not installed.

Examples LICENSE:VALIDATE? "AFG" might return LICENSE:VALIDATE "AFG",0 indicating the license is not active.

LOCk

This command enables or disables all front panel buttons and knobs. There is no front panel equivalent.

To completely disable front panel operation, combine two commands as follows: LOCK ALL; :TOUCHSCREEN:STATE OFF. To re-enable the front panel, send these two commands: LOCK NONE; :TOUCHSCREEN:STATE ON. The commands must be sent in that order.

When the front panel is locked, the front panel commands will not work and will not generate error events. You can work around a locked front panel, by using the appropriate programmatic interface commands, instead of the front-panel commands. For example, to set the trigger level to 50%, you could use TRIGGER:A SETLevel. To force a trigger, you could use TRIGGER:FORCE.

Group Miscellaneous

Syntax LOCK {ALL | NONE}
LOCK?

Related Commands UNLock TOUCHSCREEN:STATE

Arguments ALL disables all front panel controls.

NONE enables all front panel controls. This is equivalent to the UNLock ALL command.

If the instrument is in the Remote With Lockout State (RWLS), the LOCk NONE command has no effect. For more information, see the ANSI/IEEE Std 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.
Examples

LOCK ALL locks the front panel controls.

LOCK? might return :LOCK NONE, indicating that the front panel controls are enabled.

*LRN? (Query Only)

This query-only command returns the commands that list the instrument settings, allowing you to record or “learn” the current instrument settings. You can use these commands to return the instrument to the state it was in when you made the *LRN? query. This command is identical to the SET? command.

Group

Miscellaneous

Syntax

*LRN?

Related Commands

SET?

Examples

*LRN? might return the following response:

HORIZONTALSCALE;:HORIZONTAL:SAMPLERATE
6.2500E+9;:DISPLAY:WAVEVIEW1;:ZOOM:ZOOM1;:VERTICAL:SCALE
1.0000;:TRIGGER:B:TYPE EDGE;:TRIGGER:A:TYPE
EDGE;:ACQUIRE:SEQUENCE:NUMSEQUENCE
1;:ACQUIRE:SEQUENCE:MODE NUMACQS;:ACQUIRE:MODE
SAMPLE;:ACQUIRE:STOPAFTER RUNSTOP;:ACQUIRE:FASTACQ:PALETTE
TEMPERATURE;:ACQUIRE:FASTACQ:STATE 0;:ACQUIRE:NUMAVG 16T.

MAINWindow:RRBDisplaystate (No Query Form)

Sets the display state of the Results readout bar to ON (displayed) or OFF (not displayed).

Group      Miscellaneous
Syntax     MAINWindow:RRBDisplaystate {1|ON|0|OFF}
Arguments  1 or ON enables Autoset to change vertical settings.
            0 or OFF disables Autoset from changing vertical settings.
Examples   MAINWindow:RRBDisplaystate 1 expands (displays) the Results readout bar.

MATH:ADDNew (No Query Form)

This command adds the specified math.

Group      Math
Syntax     MATH:ADDNew <QString>
Related Commands   MATH:LIST?
                   MATH:DELETE
Arguments   <QString> is the quoted string specifying the math waveform to add. The argument is of the form "MATH<NR1>", where <NR1> is ≥1.
Examples    MATH:ADDNEW "MATH2" adds MATH2.
MATH:DELete (No Query Form)

This command deletes the specified math.

Group Math

Syntax MATH:DELete <QString>

Related Commands MATH:ADDNew

Arguments <QString> is a quoted string specifying the math waveform to delete. The quoted string is of the form "MATH<NR1>", where <NR1> is ≥1.

Examples MATH:DELETE "MATH1", which deletes the MATH1 waveform.

MATH:LIST? (Query Only)

This query returns a comma separated list of all currently defined math waveforms.

Group Math

Syntax MATH:LIST?

Related Commands MATH:ADDNew

Returns All currently defined math waveforms.

Examples MATH:LIST? Might return :MATH:LIST MATH1, MATH4, indicating MATH1 and MATH4 are the currently defined math waveforms.

MATH:MATH<x>:AVG:MODE

This command sets or queries the math average mode flag. If the flag is set to 1, math averaging is turned on. The math waveform is specified by x.
Group Math

Syntax MATH:MAT<h>:AVG:MODE {<NR1>|OFF|ON}

Related Commands MATH:MATH<h>:AVG:WEIGHT

Arguments

<NR1> = 0 turns off average mode, and any other integer turns on average mode. OFF turns off average mode. ON turns on average mode.

Examples MATH:MATH2:AVG:MODE ON sets the average mode on. MATH:MATH1:AVG:MODE? might return :MATH:MATH1:AVG:MODE 0, indicating average mode is off.

MATH:MATH<h>:AVG:WEIGHT

This command sets or queries the number of acquisitions at which the averaging algorithm will begin exponential averaging. The math waveform is specified by h.

Group Math

Syntax MATH:MATH<h>:AVG:WEIGHT <NR1>

Related Commands MATH:MATH<h>:AVG:MODE

Arguments <NR1> is the number of acquisitions at which the averaging algorithm will begin exponential averaging.

Examples MATH:MATH2:AVG:WEIGHT 143 sets the number of acquisitions needed to begin averaging to 20. MATH:MATH3:AVG:WEIGHT? might return :MATH:MATH3:AVG:WEIGHT 20 indicating the number of acquisitions needed to begin averaging is 8.

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer 2-413
MATH:MATH<x>:DEFine

This command allows you to define new waveforms using mathematical expressions. The query form of this command returns the math definition for the specified math waveform. The math waveform is specified by x.

You can specify a math expression from waveforms, measurements and scalar sources, functions, operands, and numerical constants.

Math expressions can be simple, such as Ch1, which specifies that a waveform should show the signal source of Channel 1 with no mathematical computation. Math expressions can also be complex, consisting of 100 plus characters and comprising many sources (including other math waveforms), functions, and operands. As an example, you can enter the expression Log(Ch1+Ch2), which specifies that the signals from channels 1 and 2 are to be algebraically added, and the base 10 log of the sum is to be shown as the final math waveform.

Group Math

Syntax MATH:MATH<x>:DEFine <QString>
MATH:MATH<x>:DEFine?

Arguments <QString> quoted string argument is the mathematical expression that defines the waveform. MATH:MATH<x>:DEFINE? is for use when the MATH:MATH<x>:TYPE is ADVANCED.

Examples MATH:MATH2:DEFine "CH1+CH2" adds the Channel 1 and Channel 2, defines the Math2 waveform to be Channel 1 and Channel 2 added together.
MATH:MATH1:DEFine? might return
MATH:MATH1:DEFine "CH2*REF2" as the expression that defines Math 1 waveform.

MATH:MATH<x>:FUNCTION

This command sets or queries the basic math arithmetic function. The math waveform is specified by x.

NOTE: This command does not affect the same Math equation in Advanced math (also accessed via the command MATH:MATH<x>:DEFINE).

Group Math
Syntax: MATH:MATH<x>:FUNCTION {ADD|SUBtract|MULTiply|DIVide}

Arguments:
- ADD sets the basic math function to add.
- SUBtract sets the basic math function to subtract.
- MULTiply sets the basic math function to multiply.
- DIVide sets the basic math function to divide.

Examples:
- MATH:MATH2:FUNCTION MULTIPLY sets the basic math function to multiply.
- MATH:MATH1:FUNCTION? might return :MATH:MATH1:FUNCTION ADD indicating the current basic math function is addition.

MATH:MATH<x>:GATING

This command specifies or returns the gating setting. It only applies to Math FFT plots. The math waveform is specified by x.

Group: Math

Syntax: MATH:MATH<x>:GATING {NONE|SCREEN|CURSor}
MATH:MATH<x>:GATING?

Related Commands: MATH:MATH<x>:TYPe

Arguments:
- NONE turns off math gating.
- SCREEN turns on gating, using the left and right edges of the screen.
- CURSor limits math to the portion of the waveform between the vertical bar cursors, even if they are off screen.

Examples:
- MATH:MATH3:GATING CURSOR sets the spectral math plot to be gated by the cursors.
- MATH:MATH2:GATING? might return :MATH:MATH:2:GATING SCREEN which indicates the spectral math plot is gated by the screen.

MATH:MATH<x>:LABel:COLor

This command sets or queries color of the specified math's label. The math waveform is specified by x.
Group Math
Syntax MATH:MATH<x>:LABEL:COLOR <QString>
Arguments <QString> is the color of the label. To return the color to the default color, send an empty string as in this example: :MATH:MATH1:LABEL:COLOR "".
Examples MATH:MATH3:LABEL:COLOR "GREEN" sets the Math 3 label color to green. MATH:MATH1:LABEL:COLOR? might return :MATH:MATH1:LABEL:COLOR "BLUE" indicating the color of the Math1 label is blue.

**MATH:MATH<x>:LABEL:FONT:BOLD**

This command sets or queries the bold state of the specified math label. The math waveform is specified by x.

Group Math
Syntax MATH:MATH<x>:LABEL:FONT:BOLD {<NR1>|OFF|ON}
Arguments <NR1> = 0 turns off bold, and any other integer turns on bold. OFF turns off bold. ON turns on bold.
Examples MATH:MATH3:FONT:BOLD ON set the math 3 label to bold. MATH:MATH2:FONT:BOLD? might return :MATH:MATH2:FONT:BOLD 0 indicating the math 2 label is not currently bold.

**MATH:MATH<x>:LABEL: FONT:ITALic**

This command sets or queries italic state of the specified math label. The math waveform is specified by x.

Group Math
Syntax MATH:MATH<x>:LABEL:FONT:ITALic {<NR1>|OFF|ON}
Arguments  

\(<\text{NR1}> = 0 \) turns off italic, and any other integer turns on italic.

\( \text{OFF} \) turns off italic.

\( \text{ON} \) turns on italic.

Examples  

\( \text{MATH:MAT}\math{x}::\text{LABel:FONT:ITALIC ON} \) set the math \( x \) label to be italic.

\( \text{MATH:MAT}\math{1}::\text{LABel:FONT:ITALIC?} \) might return \( \text{MATH:MAT}\math{1}::\text{LABel:FONT:ITALIC} 0 \) indicating the math 1 label is not currently italic.

\( \text{<NR1>} \) is the font size of the label.

\textbf{MATH:MAT}\math{x}::\text{LABel:FONT:SIZE}  

This command sets or queries font size of the specified math label. The math waveform is specified by \( x \).

Group  

Math

Syntax  

\text{MATH:MAT}\math{x}::\text{LABel:FONT:SIZE} \ <\text{NR1}>

Arguments

Examples  

\( \text{MATH:MAT}\math{4}::\text{LABel:FONT:SIZE} 32 \) sets the math 4 label size to 32.

\( \text{MATH:MAT}\math{2}::\text{LABel:FONT:SIZE?} \) might return \( \text{MATH:MAT}\math{2}::\text{LABel:FONT:SIZE} 14 \) indicating the math 2 label size is currently 14.

\textbf{MATH:MAT}\math{x}::\text{LABel:FONT:TYPE}  

This command sets or queries font type of the specified math label, such as Arial or Times New Roman. The math waveform is specified by \( x \).

Group  

Math

Syntax  

\text{MATH:MAT}\math{x}::\text{LABel:FONT:TYPE} \ <\text{QString}>

Arguments  

\(<\text{QString}> \) is the name of the font type.
Examples

MATH:MATH2:LABEL:FONT:TYPE "Serif" sets the math 2 label font type to Serif.
MATH:MATH3:LABEL:FONT:TYPE? might return
:MATH:MATH3:LABEL:FONT:TYPE "Monospace" indicating the math 3 font type is currently Monospace.

MATH:MATH<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified math label. The math waveform is specified by x.

Group Math

Syntax MATH:MATH<x>:LABel:FONT:UNDERline {<NR1>|OFF|ON}

Arguments

<NR1> = 0 turns off underline, and any other integer turns on underline.
OFF turns off underline.
ON turns on underline.

Examples

MATH:MATH3:FONT:UNDERLINE ON sets the math 3 label to be underlined.
MATH:MATH2:FONT:UNDERLINE? might return
:MATH:MATH2:FONT:UNDERLINE 0 indicating the math 2 label is currently not underlined.

MATH:MATH<x>:LABel:NAMe

This command sets or queries the label string, which is used for annotating the math waveform on the screen. The math waveform to which the label is attached is specified by x.

Group Math

Syntax MATH:MATH<x>:LABel:NAMe <QString>
MATH:MATH<x>:LABel:NAMe?

Arguments <QString> specifies the label to annotate the math waveform.
Examples

MATH:MATH2:LABel:NAMe "PROBE POINT7" assigns "Probe point7" Math 2 waveform.

MATH:MATH2:LABel:NAMe? might return :MATH:MATH2:LABel:NAMe "Probe point7", indicating that Probe point 7 is the label for the Math 2 waveform.

**MATH:MATH<x>:LABel:XPOS**

This command sets or queries the X position of the specified math label. Maths are specified by x.

**Group**
Math

**Syntax**

MATH:MATH<x>:LABel:XPOS <NR1>
MATH:MATH<x>:LABel:XPOS?

**Related Commands**
MATH:MATH<x>:LABel:YPOS

**Arguments**

<NR1> is the location (in pixels) where the label for the selected math waveform is displayed, relative to the left edge of the display.

**Examples**

MATH:MATH2:LABEL:XPOS 5 moves the waveform label for the Math 2 waveform so that it begins 5 pixels to the right of the left edge of the screen.

MATH:MATH2:LABEL:XPOS? might return :MATH:MATH2:LABEL:XPOS 2.5, indicating that the waveform label for the Math 2 waveform is currently 2.5 pixels to the right of the left edge of the display.

**MATH:MATH<x>:LABel:YPOS**

This command sets or queries the y-position of the specified math label. The Math waveform is specified by x.

**Group**
Math

**Syntax**

MATH:MATH<x>:LABel:YPOS <NR1>
MATH:MATH<x>:LABel:YPOS?

**Related Commands**
MATH:MATH<x>:LABel:XPOS
Arguments

<NR1> is the location (in pixels) where the label for the selected math waveform is displayed, relative to the baseline of the waveform.

Examples

MATH:MATH2:LABEL:YPOS -2.5 moves the waveform label for the Math 2 waveform to 2.5 pixels below the baseline of the waveform.

MATH:MATH2:LABEL:YPOS? might return :MATH:MATH2:LABEL:YPOS 0, indicating that the waveform label for the Math 2 waveform is currently located at the baseline of the waveform.

MATH:MATH<x>:SOURCE<x>

This command sets or queries the specified math source. The source in the command can be either 1 or 2. This command sets the Basic Math components in the user interface, with two sources and a function. You would also need to set the math type to Basic to see the change in the user interface but this will not effect the programmable interface. The math waveform and source are specified by x.

Group

Math

Syntax

MATH:MATH<x>:SOURCE<x> {CH<x>|MATH<x>|REF<x>}

Related Commands

MATH:MATH<x>:TYPE
MATH:MATH<x>:FUNCTION

Arguments

Arguments are possible math sources. SOURCE1 and SOURCE2 are for use when the MATH:MATH<x>:TYPE is BASIC.

Examples

MATH:MATH2:SOURCE1 CH1 sets the first source of math 2 to Channel 1.


MATH:MATH<x>:SPECTral:HORZ

This command sets or queries the horizontal display scale of the spectral math waveform. The math waveform is specified by x.

Group

Math
Syntax

`MATH:MATH<x>:SPECTral:HORZ {LOG|LINEAr}`

Arguments

- **LINEAr** sets the SpectralMag units to linear.
- **LOG** sets the SpectralMag units to log.

Examples

`MATH:MATH2:SPECTRAL:HORZ LOG` sets the horizontal display scale of the spectral math waveform to log.

`MATH:MATH3:SPECTRAL:HORZ?` might return `MATH:MATH3:SPECTRAL:HORZ LINEAR` indicating the horizontal display scale of the Math 3 spectral math waveform is currently set to linear.

**MATH:MATH<x>:SPECTral:MAG**

This command sets or queries the units of the SpectralMag function in the specified math definition string. The Math waveform is specified by x.

Group

Math

Syntax

`MATH:MATH<x>:SPECTral:MAG {LINEAr|DBM}`

`MATH:MATH<x>:SPECTral:MAG?`

Arguments

- **LINEAR** sets the SpectralMag units to linear.
- **DBM** sets the SpectralMag units to decibels. It also sets the Ref Level Offset to a value that is the equivalent of 1 mW into 50 Ω.

Examples

`MATH:MATH2:SPECTral:MAG DBM` sets the SpectralMag units for Math 2 waveform to decibels.

`MATH:MATH2:SPECTral:MAG?` might return `MATH:MATH2:SPECTral:MAG DBM`, indicating that the SpectralMag units for Math 2 waveform are set to decibels.

**MATH:MATH<x>:SPECTral:PHASE**

This command sets or queries the units of a SpectralPhase function in the specified math definition string. The Math waveform is specified by x.

Group

Math
**Commands listed in alphabetical order**

**Syntax**

MATH:MATH<x>:SPECTral:PHASE {DEGREES|RADians|GROUPDelay}

MATH:MATH<x>:SPECTral:PHASE?

**Arguments**

DEGREES sets the SpectralPhase units to degrees.

RADIANS sets the SpectralPhase units to radians.

GROUPDELAY sets the SpectralPhase units to groupdelay, which computes the derivative of unwrapped phase spectrum. Units are expressed in seconds.

**Examples**

MATH:MATH2:SPECTral:PHASE DEGREES sets the SpectralPhase units for Math 2 waveform to degrees.

MATH:MATH2:SPECTral:PHASE? might return :MATH:MATH2:SPECTral:PHASE RADIANS, indicating that the SpectralPhase units for Math 2 waveform are set to radians.

**MATH:MATH<x>:SPECTral:SOUrce**

This command sets or queries the specified spectral math source. This only works with a math of type FFT. The math waveform is specified by x.

**Group**

Math

**Syntax**

MATH:MATH<x>:SPECTral:SOURce {CH<x>|MATH<x>|REF<x>}

MATH:MATH<x>:SPECTral:SOURce?

**Related Commands**

MATH:MATH<x>:TYPe

**Arguments**

Arguments are math sources. MATH:MATH<x>:SPECTRAL:SOURCE is for use when the MATH:MATH<x>:TYPE is FFT.

**Examples**

MATH:MATH3:SPECTRAL:SOURCE REF3 sets the source of the Spectral Math waveform to Ref 3.

MATH:MATH<x>:SPECTral:SUPPress

This command sets or queries whether suppression threshold for the specified math waveform is enabled. This is only applied when Spectral Plot type is Phase. The math waveform is specified by x.

Group Math

Syntax MATH:MATH<x>:SPECTral:SUPPress {OFF|ON|0|1} MATH:MATH<x>:SPECTral:SUPPress?

Related Commands MATH:MATH<x>:SPECTral:TYPE

Arguments 0 disables suppression threshold for the specified math waveform. 1 enables suppression threshold for the specified math waveform. ON enables suppression threshold for the specified math waveform. OFF disables suppression threshold for the specified math waveform.


MATH:MATH<x>:SPECTral:SUPPress:VALue

This command sets or queries in volts the value of suppression threshold of the specified math waveform. This requires the Spectral type to be Phase and the Suppression to be enabled for this PI command to have any affect. The math waveform is specified by x.

Group Math


Related Commands MATH:MATH<x>:SPECTral:TYPE MATH:MATH<x>:SPECTral:SUPPress
Arguments

<NR3> is the value of suppression threshold of the specified math waveform in volts.

Examples

MATH:MATH3:SPECTRAL:SUPPRESS:VALUE 100.0E-3 sets the suppression threshold of Math 3 to 100 mV.
MATH:MATH1:SPECTRAL:SUPPRESS:VALUE? might return MATH:MATH1:SPECTRAL:SUPPRESS:VALUE 10.0000E+3 indicating the suppression threshold of Math 1 is currently set to 10 kV.

MATH:MATH<x>:SPECTral:TYPE

This command sets or queries the FFT type selected for spectral analysis. The math waveform is specified by x.

Group  Math

Syntax  MATH:MATH<x>:SPECTral:TYPE {MAGNitude|PHASe|REAL|IMAGinary}
MATH:MATH<x>:SPECTral:TYPE?

Arguments

MAGNitude specifies the magnitude spectral function.
PHASe specifies the phase spectral function.
REAL specifies the real spectral function.
IMAGinary specifies the imaginary spectral function.

Examples

MATH:MATH1:SPECTral:TYPE REAL specifies the real spectral function.
MATH:MATH1:SPECTral:TYPE? might return :MATH:MATH1:SPECTRAL:TYPE MAGNITUDE indicating the math is the magnitude spectral function.

MATH:MATH<x>:SPECTral:UNWRap

This command sets or queries whether phase unwrap of the spectral analyzer output data is enabled. The Math waveform is specified by x.

Group  Math

Syntax  MATH:MATH<x>:SPECTral:UNWRap {OFF|ON|0|1}
MATH:MATH<x>:SPECTral:UNWRap?
Arguments
0 disables phase unwrap for the specified math waveform.
1 enables phase unwrap for the specified math waveform.
ON enables phase unwrap for the specified math waveform.
OFF disables phase unwrap for the specified math waveform.

Examples
MATH1:SPECTRAL:UNWRAP ON enables phase unwrap of the spectral analyzer output data.
MATH1:SPECTRAL:UNWRAP? might return :MATH1:SPECTRAL:UNWRAP 0, indicating that the phase unwrap of the spectral analyzer output data is disabled.

MATH:MATH<x>:SPECTral:UNWRap:DEGrees
This command sets or queries how many degrees adjacent phase values can jump before being unwrapped. This requires the Spectral type to be Phase and the UNWRAP to be enabled for this PI command to have any affect. The math waveform is specified by x.

Group
Math

Syntax
MATH:MATH<x>:SPECTral:UNWRap:DEGrees <NR3>
MATH:MATH<x>:SPECTral:UNWRap:DEGrees?

Related Commands
MATH:MATH<x>:SPECTral:TYPE
MATH:MATH<x>:SPECTral:UNWRap

Arguments
<NR3> is the value of unwrap phase in degrees.

Examples
MATH:MATH2:SPECTRAL:UNWRAP:DEGREES 90 sets the unwrap phase of the spectral Math2 to 90 degrees.

MATH:MATH<x>:SPECTral:WINdow
This command sets or queries the window function used to apply the specified FFT window to the input data for the specified math waveform. The Math waveform is specified by x. A spectral window determines what the filter shape
of the spectral analyzer will be in the frequency domain. It can be described by a mathematical function that is multiplied point-by-point times the input data to the spectral analyzer.

Following is a list of arguments that specify the window function used to multiply the input data. The windows are listed in the order of their ability to resolve frequencies (resolution bandwidth).

Group                  Math

Syntax                 MATH:MATH<x>:SPECTral:WINdow {RECTANGular|HAMMing|HANNing|BLACKMANHarris|KAISERBessel|GAUSSian|FLATTOP2|TEKEXPonential}
MATH:MATH<x>:SPECTral:WINdow?

Related Commands      MATH:MATH<x>:TYPe

Arguments             RECTANGular window function is equivalent to multiplying all gate data by one.
HAMMing window function is based on a cosine series.
HANNing window function is based on a cosine series.
BLACKMANHarris window function is based on a cosine series.
KAISERBessel window function is based on a cosine series.
GAUSSian window function has the best localization characteristics in the joint time/frequency plane.
FLATTOP2 window function is a cosine series window with a flattened frequency response lobe.
TEKEXPonential window has an exponential nonsymmetrical shape in the time domain and a triangular shape in the frequency domain.

Examples              MATH2:SPECTRAL:WINDOW GAUSSIAN applies a Gaussian window to the spectral analyzer input data.
MATH2:SPECTRAL:WINDOW? might return :MATH2:SPECTRAL:WINDOW BLACKMANHarris, indicating that the window function used to multiply the spectral analyzer input data is the BLACKMANHarris function.

MATH:MATH<x>:TYPe

This command sets or queries the math type. The math waveform is specified by x.
Commands listed in alphabetical order

Group    Math

Syntax    MATH:MATH<x>:TYPE {BASic|FFT|ADVanced}

Arguments

BASic  set the type to basic math.

FFT  sets the type to FFT math, which can use any live analog or reference waveform in the time domain. NOTE. You can also use FFT as part of a math expression by declaring the type.

ADVanced  See examples for the command MATH:MATH<x>:DEFine.

ADVanced sets the type to advanced math.

Examples

MATH:MATH2:TYPE BASIC sets the type of Math 2 to basic.

MATH:MATH2:TYPE? might return :MATH:MATH2:TYPE FFT indicating the type of Math 4 is currently FFT.

MATH:MATH<x>:VUNIT

This command specifies or returns the math custom vertical units. The math waveform is specified by x.

Group    Math

Syntax    MATH:MATH<x>:VUNIT <QString>

Arguments  <QString>  is the custom vertical units.

Examples

MATH:MATH2:VUNIT "Small"

MATH:MATH2:VUNIT? might return :MATH:MATH2:VUNIT "Large" indicating the Math 2 vertical unit is set to "Large".

MATHArbflt<x>:FILEpath

This command or query sets the file path for a file of filter coefficients for the specified arbitrary filter. Setting a path will read that file and load the filter for ARBFLT<x>. Access these filters using a Math with an expression of the form "ARBFlt<x>0".
### Commands listed in alphabetical order

**Group**: Math

**Syntax**

```
MATHArbflt<x>:FILEpath <QString>
MATHArbflt<x>:FILEpath?
```

**Arguments**

`<QString>` specifies the path to the file of filter coefficients.

**Examples**

```
MATHARBFLT8:FILEPATH "c:/myfilters/20mhz.flt" reads filter coefficients from the file and loads the filter for ARBFLT8.

MATHARBFLT1:FILEPATH? might return :MATHARBFLT1:FILEPATH "C:\Users\Public\Tektronix\TekScope\Math Arbitrary Filters\LowPass-Norm\lowpass_0.bw.flt", indicating that the path to the file of filter coefficients is set to "C:\Users\Public\Tektronix\TekScope\Math Arbitrary Filters\LowPass-Norm\lowpass_0 bw.flt".
```

### MEASUrement? (Query Only)

This query-only command returns all measurement parameters.

**Group**: Measurement

**Syntax**

```
MEASUrement?
```

**Examples**

```
MEASUREMENT? might return :
MEASUREMENT:GATING OFF; IMMED:TYPE UNDEFINED; UNITS "V"; SOURCE1 CH1; SOURCE2 CH1; SOURCE1:SIGTYPE PULSE; MEASUREMENT:IMMED:SOURCE2:SIGTYPE PULSE; MEASUREMENT:IMMED:DELAY:EDGE1 RISE; EDGE2 RISE; DIRECTION FORWARDS; MEASUREMENT:IMMED:REFLEVEL:METHOD PERCENT; ABSOLUTE:HIGH 0.0000; LOW 0.0000; MID1 0.0000; MID2 0.0000; MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH 90.0000; LOW 10.0000; MID1 50.0000; MID2 50.0000; MEASUREMENT:IMMED:METHOD HISTOGRAM; NOISE HIGH; MEASUREMENT:MEAS1:STATE 0; TYPE UNDEFINED; UNITS "V"; SOURCE1 CH1; SOURCE2 CH1; SOURCE1:SIGTYPE PULSE; MEASUREMENT:MEAS1:SOURCE2:SIGTYPE PULSE; MEASUREMENT:MEAS1:DELAY:EDGE1 RISE; EDGE2 RISE; DIRECTION FORWARDS; MEASUREMENT:MEAS1:REFLEVEL:METHOD PERCENT; ABSOLUTE:HIGH 0.0000; LOW 0.0000; MID1 0.0000; MID2 0.0000; MEASUREMENT:MEAS1:REFLEVEL:PERCENT:HIGH 90.0000; LOW 10.0000; MID1 50.0000; MID2 50.0000; MEASUREMENT:MEAS1:METHOD HISTOGRAM; NOISE HIGH; MEASUREMENT:MEAS2:STATE 0; TYPE UNDEFINED; UNITS "V"; SOURCE1 CH1; SOURCE2
```
MEASUrement:ADDMEAS (No Query Form)

This command adds a measurement.

Group    Measurement

Syntax

MEASUrement:ADDMEAS {ACCOMMONMODE|ACRMS|AMPLITUDE|AREA|BASE|BITAMPLITUDE|BITHIGH|BITLOW|BURSTWIDTH|COMMONMODE|DATARATE|DCD|DJ|DDRALS|DDRSPAN|DDRTCHABS|DDRTCHAVERAGE|DDRTCKAVERAGE|DDRTCLABS|DDRTCLAVERAGE|DDRTERRM|DDRTERRN|DDRTJITCC|DDRTJITDUTY|DDRTJITPER|DDRTPST|DDRTSPREP|DDRTSPREP|DDRTWXAC|DDRTDQSCK|DELAY|DJ|DJC|FALLSLEWRATE|FALLTIME|FREQUENCY|F2|F4|F8|HEIGHT|HEIGHTBER|HIGH|HIGHTIME|HOLD|JITTERSUMMARY|J2|J9|LOW|LOWTIME|MADJ|MEAN|MINIMUM|NDUTY|NOVERSHOOT|NPERIOD|NPJ|NWIDTH|PDDUTY|PERIOD|PHASE|PHASENOISE|PJ|PK2PK|POVERSHOOT|PWIDTH|QFACTOR|RISESLEWRATE|RISETIME|RJ|RJDIRAC|RMS|SETUP|SKEW|SRJ|SSCFREQDEV|SSCMODRATE|TIE|TIMEOUTSIDELEVEL|TJBER|TNTRATIO|TOP|UNITINTERVAL|VDIFFXOVR|WIDTH|WIDTHBER}

Arguments

ACCOMMONMODE AC Common Mode (Pk-Pk) is the peak-to-peak of the common mode voltage of two sources. This measurement is made across the entire record.

ACRMS (AC RMS) is the true Root Mean Square of the data points, about the Mean. This measurement can be made across the entire record, or on each cycle in the record.

AMPLITUDE is the difference between the Top value and the Base value. This measurement can be made across the entire record, or on each cycle in the record.

Amplitude = High - Low

AREA is the area under the curve, calculated by integrating the data points. The area measured above ground is positive. The area measured below ground is negative. This measurement can be made across the entire record, or on each cycle in the record.

BASE is the most common data value below the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

BITAMPLITUDE (Bit Amplitude) is the difference between the amplitudes of the 1 bit and the 0 bit surrounding a transition. The amplitude is measured over a user
specified portion at the center of the recovered unit interval. This measurement is made on each transition bit in the record (Mean) or across the entire record (Mode).

**BITHIGH** (Bit High) is the amplitude of a 1 bit. The amplitude is measured over a user-specified portion at the center of the recovered unit interval. This measurement is made on each high bit in the record (Mean) or across the entire record (Mode).

**BITLOW** (Bit Low) is the amplitude of a 0 bit. The amplitude is measured over a user-specified portion at the center of the recovered unit interval. This measurement is made on each high bit in the record (Mean) or across the entire record (Mode).

**BURSTWIDTH** (Burst Width) is the duration of a series of adjacent crossings of the Mid reference level (RM). Bursts are separated by a user-defined idle time (tI). This measurement is made on each burst in the record.

**COMMONMODE** (DC Common Mode) is the arithmetic mean of the common mode voltage of two sources. This measurement is made across the entire record.

**DATARATE** (Data Rate) is the reciprocal of Unit Interval. This measurement is made on each bit in the record.

**DCD** (duty cycle distortion) is the peak-to-peak amplitude of the component of the deterministic jitter correlated with the signal polarity. This measurement is made across the entire record.

**DDJ** (data dependent jitter) is the peak-to-peak amplitude of the component of the deterministic jitter correlated with the data pattern in the waveform. This measurement is made across the entire record.

**DDRAOS** (area above signal) is the total area of the signal above a specified reference level. This measurement is made across the entire record.

**DDRAOSSPERCK** (area over signal for tCK events) is the total area of the signal above a specified reference level calculated over consecutive tCK intervals. It is applicable to clock and address/command waveforms.

**DDRAOSPERUI** (area over signal for UI events) is the total area of the signal above a specified reference level calculated over consecutive unit intervals. It is applicable to data and data strobe waveforms.

**DDRAUS** (area under signal) is the total area of the signal below a specified reference level. This measurement is made across the entire record.

**DDRAUSPERCK** (area under signal for tCK events) is the total area of the signal below a specified reference level calculated over consecutive tCK intervals. It is applicable to clock and address/command waveforms.

**DDRAUSPERUI** (area under signal for UI events) is the total area of the signal below a specified reference level calculated over consecutive unit intervals. It is applicable to data and data strobe waveforms.
DDRHOLDDIFF (hold difference) is the elapsed time between the specified edge of a single-ended clock waveform and the specified edge of a differential data waveform. The measurement uses the closest respective waveform edges that fall within the range limits. This measurement is made across the entire record.

DDRSETUPDIFF (setup difference) is the elapsed time between the specified edge of a single-ended clock waveform and when the specified edge of a differential data waveform crosses a specified level. The measurement uses the closest respective waveform edges that fall within the range limits. This measurement is made across the entire record.

DDRTCHABS (absolute high pulse width) is the absolute value of the high pulse width as measured from one rising edge to the next falling edge.

DDRTCHAVERAGE (average high pulse width) is the average value of the high pulse width as measured from one rising edge to the next falling edge, across 200 consecutive cycles. This measurement is made across the entire record.

DDRTCKAVERAGE (average clock period) is the average clock period calculated from rising edge to rising edge, across 200 consecutive cycles. This measurement is made across the entire record.

DDRTCLAVERAGE (average low pulse width) is the average value of the low pulse width as measured from one falling edge to the next rising edge, across 200 consecutive cycles.

DDRTERRMN (cumulative error) is the cumulative error across multiple consecutive defined cycles from tCK(avg).

DDRTERRN (cumulative error) is the cumulative error across specified consecutive cycles from tCK(avg). In other words, this measures the time difference between the sum of the clock period from a 200 cycle window and n times tCK(avg).

DDRTJITCC (cycle to cycle jitter period) is the absolute difference in clock period between two consecutive clock cycles. This measurement is made across the entire record.

DDRTJITDUTY (half period jitter) is the largest elapsed time between tCH and tCH(avg), and tCL and tCL(avg), over 200 consecutive cycles.

DDRTJITPER (clock period jitter) is the largest deviation of any tCK signal from tCK(avg). This measurement is made across the entire record.

DDRTPSST (read/write burst postamble) is the width of the Read or Write burst postamble, measured from the last falling edge of the mid reference level to the start of an undriven state. This measurement is made across the entire record.
Commands listed in alphabetical order

DDTRPRE (read burst preamble) is the width of the Read burst preamble, measured from exiting tristate levels to the first driving edge of the differential strobe. This measurement is made across the entire record.

DDRWPRE is the width of the Write burst preamble, measured from exiting tristate levels to the first driving edge of the differential strobe. This measurement is made across the entire record.

DDRVI
XAC is the differential input cross-point voltage measured from the true state transition (and its complement) to a specified reference level, measured on a single-ended signal.

DDRTDQSCK is the strobe output access time, measured between the rising edge of the clock and before or after the differential strobe Read preamble time. Signal edges are determined by the mid-ref threshold level settings.

DELay is the time between the specified Mid reference level (RM) crossing on one source to a specified Mid reference level (RM) crossing on a second source. This measurement is made on the first occurrence in the record.

DJ (deterministic jitter) is the peak-to-peak amplitude of all timing errors that exhibit deterministic behavior. This measurement is made across the entire record.

DJDIRAC (dual-dirac deterministic jitter) is deterministic jitter based on a simplifying assumption that the histogram of all deterministic jitter can be modeled as a pair of equal-magnitude Dirac functions. This measurement is made across the entire record.

EYEHIGH (Eye High) is the amplitude of a high (1) bit measured at a user specified location within the recovered unit interval. This measurement is made on each high bit in the record.

EYELOW (Eye Low) is the amplitude of a low (0) bit measured at a user specified location within the recovered unit interval. This measurement is made on each low bit in the record.

FALLSLEWRATE (Falling Slew Rate) is the rate of change in voltage as an edge transitions from the Top reference level (RT) to the Bottom reference level (RB). This measurement is made on each cycle in the record.

FALLTIME (Fall Time) is the time required for an edge to fall from the Top reference level (RT) to the Base reference level (RB). This measurement is made on each cycle in the record.

FREQuency is the reciprocal of Period. This measurement is made on each cycle in the record.

F2 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 2. This measurement is made across the entire record.

F4 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 4. This measurement is made across the entire record.
F8 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 8. This measurement is made across the entire record.

HIGH (Eye High) is the amplitude of a high (1) bit measured at a user specified location within the recovered unit interval. This measurement is made on each high bit in the record.

HEIGHT (Eye Height) is the minimum vertical eye opening at the center of the recovered unit interval. This measurement is made across the entire record.

HEIGHTBER (Eye Height@BER) is the predicted vertical eye opening that will be violated with a probability equal to the bit error rate. This measurement is made across the entire record.

HIGH

HIGHTIME (High Time) is the time the signal remains above the Top reference level (RT). This measurement is made on each cycle in the record.

HOLD (Hold Time) is the time between the specified Mid reference level crossing (RM) on the Clock source to the closest specified Mid reference level (RM) crossing on the Data source. This measurement is made on each specified Clock edge in the record.

JITTERSUMMARY (Jitter Summary) is a group consisting of the following measurements: TIE, TJ@BER, Eye Width@BER, Eye Height@BER, RJ-δδ, DJ-δδ, PJ, DDJ, DcD, F/2, F/4, F/8.

J2 is the total jitter at a bit error rate of 2.5e-3 (TJ@2.5e-3). This measurement is made across the entire record.

J9 is the total jitter at a bit error rate of 2.5e-10 (TJ@2.5e-10). This measurement is made across the entire record.

LOW (Eye Low) is the amplitude of a low (0) bit measured at a user specified location within the recovered unit interval. This measurement is made on each low bit in the record.

LOWTIME (Low Time) is the time the signal remains below the Base reference level (RB). This measurement is made on each cycle in the record.

MAXimum is the maximum data point. This measurement can be made across the entire record, or on each cycle in the record.

MEAN is the arithmetic mean of the data points. This measurement can be made across the entire record, or on each cycle in the record.

MINimum is the minimum data point. This measurement can be made across the entire record, or on each cycle in the record.

NDuty (Negative Duty Cycle) is the ratio of the Negative Pulse Width to the Period. This measurement is made on each cycle in the record.

$$\text{Negative Duty Cycle} = \frac{\text{Negative Width}}{\text{Period}} \times 100\%$$
NPERIOD (Duration N-Periods) is the time required to complete N cycles. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

NPJ (non-periodic jitter) is the portion of the BUJ (bounded uncorrelated jitter) that is random. BUJ excludes DDJ, DCD and RJ. This measurement is made across the entire record.

NOvershoot (Negative Overshoot) is the difference between Minimum and Base, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

\[ \text{Negative Overshoot} = \frac{\text{Base} - \text{Minimum}}{\text{Amplitude}} \times 100\% \]

NWIdth (Negative Pulse Width) is the time the signal remains below the Mid reference level (RM). This measurement is made on each cycle in the record.

PDUTY (Positive Duty Cycle) is the ratio of the Positive Pulse Width to the Period. This measurement is made on each cycle in the record.

\[ \text{Positive Duty Cycle} = \frac{\text{Positive Width}}{\text{Period}} \times 100\% \]

PERIOD is the time required to complete a cycle. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

PHASE is the ratio of the Skew between two sources to the Period of the first source. This measurement is made on each cycle in the record.

PHASENOISE (Phase Noise) is the RMS magnitude of all integrated jitter falling within a user specified offset range of the fundamental clock frequency. This measurement is made across the entire record.

PJ (periodic jitter) is the peak-to-peak amplitude of the uncorrelated sinusoidal components of the deterministic jitter. This measurement is made across the entire record.

PK2Pk (Peak-to-peak) is the difference between Maximum and Minimum. This measurement can be made across the entire record, or on each cycle in the record.

POvershoot (Positive Overshoot) is the difference between Maximum and Top, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

\[ \text{Positive Overshoot} = \frac{\text{Maximum} - \text{Top}}{\text{Amplitude}} \times 100\% \]

PWIDTH (Positive Pulse Width) is the time the signal remains above the Mid reference level (RM). This measurement is made on each cycle in the record.

QFACTOR (Q-Factor) is the ratio of the vertical eye opening to RMS vertical noise measured at a user specified location within the recovered unit interval. This measurement is made across the entire record.
RISESLEWRATE (Rising Slew Rate) is the rate of change in voltage as an edge transitions from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RISETIME Rise Time is the time required for an edge to rise from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RJ (random jitter) is the RMS magnitude of all random timing errors following a Gaussian distribution. This measurement is made across the entire record.

RJDIRAC (dual-dirac random jitter) is random jitter based on a simplifying assumption that the histogram of all deterministic jitter can be modeled as a pair of equal-magnitude Dirac functions. This measurement is made across the entire record.

RMS is the true Root Mean Square of the data points. This measurement can be made across the entire record, or on each cycle in the record.

SRJ (sub-rate jitter) is the composite jitter due to periodic components at 1/2, 1/4 and 1/8 of the data rate. This measurement is made across the entire record.

SSCFREQDEV (SSC Frequency Deviation) is the spread spectrum clock frequency deviation. This measurement enables a time trend plot of the spread spectrum clock modulation profile. This measurement is made on each cycle in the record.

SSCMODRATE (SSC Modulation Rate) is the modulating frequency of a spread spectrum clock. This measurement is made on each cycle in the record.

SETUP (Setup Time) is the time between the specified Mid reference level (RM) crossing on the Data source to the closest specified Mid reference level (RM) crossing on the Clock source. This measurement is made on each specified Clock edge in the record.

SKEW Skew is the time between the specified Mid reference level (RM) crossing on one source to the following specified Mid reference level (RM) crossing on a second source. This measurement is made on each cycle in the record.

TIE (time interval error) is the difference, in time, between an edge in the source waveform and the corresponding edge in a recovered reference clock. This measurement is made on each edge in the waveform.

TIMEOUTSIDELEVEL Time Outside Level is the time the signal remains above the Top reference level (RT) and/or below the Base reference level (RB). This measurement is made on each occurrence in the record.

TJBER (total jitter at a specified bit error rate) is the predicted peak-to-peak amplitude of jitter that will only be exceeded with a probability equal to the bit error rate. This measurement is made across the entire record.

TNTRATIO T/nT Ratio is the ratio of a non-transition bit voltage (2nd and subsequent bit voltage after a transition) to its nearest preceding transition bit voltage (1st bit voltage after the transition). Bit voltages are measured at the
interpolated midpoint of the recovered unit interval. This measurement is made on each non-transition bit in the record.

**TOP** is the most common data value above the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

**UNITINTERVAL** (Unit Interval) is the time difference between two successive bits. This measurement is made on each bit in the record.

**VDIFFXOVR** (Differential Crossover) is the voltage level of a differential signal pair at the crossover points. This measurement is made at each crossover point in the record.

**WIDTH** (Eye Width) is the minimum horizontal eye opening at the user specified reference level. This measurement is made across the entire record.

**WIDTHBER** (Eye Width@BER) is the predicted horizontal eye opening that will be violated with a probability equal to the bit error rate. This measurement is made across the entire record.

**Examples**

```
MEASUREMENT:ADDMEAS FREQUENCY adds a frequency measurement.

MEASUREMENT:ADDNew (No Query Form)
```

This command adds the specified measurement.

**Group** Measurement

**Syntax** MEASUREMENT:ADDNew "QString"

**Arguments** "QString" is the measurement to add. The argument is of the form "MEAS<NR1>" where NR1 ≥ 1.

**Examples** MEASUREMENT:ADDNew "MEAS11" adds measurement 11.

**MEASUREMENT:ANNOTate**

This command sets or queries the annotation state for measurements.

**Group** Measurement

2-436 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
Commands listed in alphabetical order

Syntax

**MEASUrement:ANNOTate** {OFF|AUTO}

MEASUrement:ANNOTate?

**Arguments**

OFF turns off measurement annotations.

AUTO turns on visible measurement annotations.

**Examples**

MEASUREMENT:ANNOTATE OFF turns off measurement annotations.

MEASUREMENT:ANNOTATE? might return :MEASUREMENT:ANNOTATION:STATE OFF, indicating that no measurement annotations are active.

**MEASUrement:AUTOset (No Query Form)**

This command performs an analysis jitter autoset.

**Group**

Measurement

**Syntax**

MEASUrement:AUTOset EXECute

**Examples**

MEASUREMENT:AUTOSET EXECute performs an analysis jitter autoset.

**MEASUrement:CH<x>:REFLevels:ABSolute:FALLHigh**

This command sets or queries the value used as the high reference level of the falling edge when the source ref level method is set to absolute. The channel number is specified by x.

**Group**

Measurement

**Syntax**

MEASUrement:CH<x>:REFLevels:ABSolute:FALLHigh <NR3>

MEASUREMENT:CH<x>:REFLevels:ABSolute:FALLHigh?

**Arguments**

<NR3> is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHod is set to Absolute.

**Examples**

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLHIGH 1.5 sets the high reference level of the falling edge to 1.5 V.
MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLHIGH?

might return
:MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000 indicating
that the high reference level of the falling edge is 1.0 V.

**MEASUREMENT:CH<x>:REFLevels:ABSolute:FALLLow**

This command sets or queries the value used as the low reference level of the
falling edge when the source ref level method is set to absolute. The channel
number is specified by x.

**Group**
Measurement

**Syntax**
MEASUREMENT:CH<x>:REFLevels:ABSolute:FALLLow <NR3>
MEASUREMENT:CH<x>:REFLevels:ABSolute:FALLLow?

**Arguments**
<NR3> is the high reference level, and is the zero percent level when
MEASUREMENT:IMMed:REFLevel:METHod is set to Absolute.

**Examples**
MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLLOW 1.5 sets the low
reference level of the falling edge to 1.5 V.

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLLOW? might return
:MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLLOW 1.5000 indicating
that the low reference level of the falling edge is 1.5 V.

**MEASUREMENT:CH<x>:REFLevels:ABSolute:FALLMid**

This command sets or queries the value used as the mid reference level of the
falling edge when the source ref level method is set to absolute. The channel
number is specified by x.

**Group**
Measurement

**Syntax**
MEASUREMENT:CH<x>:REFLevels:ABSolute:FALLMid <NR3>
MEASUREMENT:CH<x>:REFLevels:ABSolute:FALLMid?

**Arguments**
<NR3> is the mid reference level used to calculate the mid reference level when
the measurement's Ref level method is set to Absolute.
MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:FALLMID

Sets the mid reference level of the falling edge to 0.0 V.

MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:FALLMID? might return
MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0 indicating
that the mid reference level of the falling edge is 0.0 V.

MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:HYSTERESIS

This command sets or queries the value of the hysteresis of the reference level when the source ref level method is set to absolute. The channel number is specified by x.

Syntax
MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:HYSTERESIS <NR3>
MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:HYSTERESIS?

Arguments
<NR3> is the hysteresis value used for autoset.

Examples
MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3 sets
the reference hysteresis level to 30 mV.

:MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3 indicating that reference hysteresis level is set to 30 mV.

MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:RISEHigh

This command sets or queries the value used as the high reference level of the rising edge when the source ref level method is set to absolute. The channel number is specified by x.

Syntax
MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:RISEHigh <NR3>
MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:RISEHigh?

Arguments
<NR3> is the high reference level of the rising edge when the source ref level method is set to absolute.
Examples

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEHIGH 1.5 sets the high reference level of the rising edge to 1.5 V.

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEHIGH? might return :MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000 indicating that the high reference level of the rising edge is 1.0 V.

MEASUREMENT:CH<x>:REFLevels:ABSolute:RISELow

This command sets or queries the value used as the low reference level of the rising edge when the source ref level method is set to absolute. The channel number is specified by x.

Group Measurement

Syntax

MEASUREMENT:CH<x>:REFLevels:ABSolute:RISELow <NR3>

MEASUREMENT:CH<x>:REFLevels:ABSolute:RISELow?

Arguments

<NR3> is the low reference level of the rising edge when the source ref level method is set to absolute.

Examples

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISELOW 1.5 sets the low reference level of the rising edge to 1.5 V.

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISELOW? might return :MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISELOW 1.0000 indicating that the low reference level of the rising edge is 1.0 V.

MEASUREMENT:CH<x>:REFLevels:ABSolute:RISEMid

This command sets or queries the value used as the mid reference level of the rising edge when the source ref level method is set to absolute. The channel number is specified by x.

Group Measurement

Syntax

MEASUREMENT:CH<x>:REFLevels:ABSolute:RISEMid <NR3>

MEASUREMENT:CH<x>:REFLevels:ABSolute:RISEMid?

Arguments

<NR3> is the mid reference level of the rising edge when the source ref level method is set to absolute.
MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEMID 0 sets the mid reference level of the rising edge to 0.0 V.

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEMID? might return :MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0 indicating that the mid reference level of the rising edge is 0.0 V.

MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:TYPE

This command sets or queries the reference level type for the source. The channel number is specified by x.

Syntax

MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:TYPE {SAME|UNIQUE}
MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:TYPE?

Arguments

SAME specifies that the absolute reference levels for the specified measurement channel are the same.

UNIQUE specifies that the absolute reference levels for the specified measurement channel are not the same.

Examples

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:TYPE SAME set the reference levels to be the same.


MEASUREMENT:CH<x>:REFLEVELS:BASETop

This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement. The channel number is specified by x.

Syntax

MEASUREMENT:CH<x>:REFLEVELS:BASETop {AUTO|MINMax|MEANhistogram|MODEhistogram|EYEhistogram}
MEASUREMENT:CH<x>:REFLEVELS:BASETop?
Arguments

AUTO automatically chooses a reference level method.

MINMax specifies that reference levels are relative to the measurement MIN and MAX.

MEAN histogram specifies that reference levels are relative to the histogram mean BASE and TOP.

MODE histogram specifies that reference levels are relative to the histogram mode BASE and TOP.

EYE histogram specifies that reference levels are relative to the eye histogram BASE and TOP.

Examples

MEASUREMENT:CH2:REFLEVELS:BASETOP MINMAX specifies that reference levels are relative to the measurement MIN and MAX.

MEASUREMENT:CH2:REFLEVELS:BASETOP? might return
:MEASUREMENT:CH2:REFLEVELS:BASETOP AUTO indicating the reference levels are chosen automatically.

MEASUrement:CH<x>:REFLevels:METHod

This command sets or queries the method used to calculate reference levels for the measurement. The channel number is specified by x.

Group Measurement

Syntax MEASUrement:CH<x>:REFLevels:METHod {PERCent|ABSolute}
MEASUrement:CH<x>:REFLevels:METHod?

Arguments

PERCent specifies percent reference level units.

ABSolute specifies absolute reference level units.

Examples

MEASUREMENT:CH2:REFLEVELS:METHod PERCENT sets reference levels to be calculated in percent.

MEASUREMENT:CH2:REFLEVELS:METHod? might return
:MEASUREMENT:CH2:REFLEVELS:METHod PERCENT indicating the reference levels are calculated in percent.
MEASUrement:CH<x>:REFLevels:PERCent:FALLHigh

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the source ref level method is set to percent. The channel number is specified by x.

Group Measurement

Syntax MEASUREMENT:CH<x>:REFLevels:PERCENT:FALLHigh <NR3>
MEASUREMENT:CH<x>:REFLevels:PERCent:FALLHigh?

Arguments <NR3> is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

Examples MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLHIGH 90.0000 sets the high reference level for the falling edge to 90%.

:MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLHIGH 90.0000 indicating the high reference level for the falling edge is set to 90%.

MEASUrement:CH<x>:REFLevels:PERCent:FALLLow

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the source ref level method is set to percent. The channel number is specified by x.

Group Measurement

Syntax MEASUREMENT:CH<x>:REFLevels:PERCENT:FALLLow <NR3>
MEASUREMENT:CH<x>:REFLevels:PERCent:FALLLow?

Arguments <NR3> is the percentage (where 100% is equal to TOP) used to calculate the low reference level when the measurement Ref level method is set to Percent.

Examples MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLLOW 10.0000 sets the reference levels for the falling edge to 10%.

**MEASUrement:CH<x>:REFLevels:PERCent:FALLMid**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent. The channel number is specified by x.

**Group** Measurement

**Syntax**

- MEASUREMENT:CH<x>:REFLevels:PERCent:FALLMid <NR3>
- MEASUREMENT:CH<x>:REFLevels:PERCent:FALLMid?

**Arguments**

<NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

**Examples**

- MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLMid 50.0000 sets the MID reference level for the falling edge to 50%.

**MEASUrement:CH<x>:REFLevels:PERCent:HYSTeresis**

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement ref level method is set to percent. The channel number is specified by x.

**Group** Measurement

**Syntax**

- MEASUREMENT:CH<x>:REFLevels:PERCent:HYSTeresis <NR3>
- MEASUREMENT:CH<x>:REFLevels:PERCent:HYSTeresis?

**Arguments**

<NR3> is the hysteresis value used for the autoset.
Examples

MEASUREMENT:CH2:REFLEVELS:PERCENT:HYSTERESIS 5.0000 sets the reference level hysteresis to 5.0 mV.

MEASUREMENT:CH2:REFLEVELS:PERCENT:HYSTERESIS? might return :MEASUREMENT:CH2:REFLEVELS:PERCENT:HYSTERESIS 5.0000 indicating the reference level hysteresis is set to 5.0 mV.

**MEASUrement:CH<x>:REFLevels:PERCent:RISEHigh**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement ref level method is set to percent. The channel number is specified by x.

Group Measurement

Syntax MEASUrement:CH<x>:REFLevels:PERCent:RISEHigh <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:RISEHigh?

Arguments <NR3> is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

Examples

MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEHIGH 90.0000 sets the high reference level for the rising edge to 90%.

MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEHIGH? might return :MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEHIGH 90.0000 indicating the high reference level for the rising edge is set to 90%.

**MEASUrement:CH<x>:REFLevels:PERCent:RISELow**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement ref level method is set to percent. The channel number is specified by x.

Group Measurement

Syntax MEASUrement:CH<x>:REFLevels:PERCent:RISELow <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:RISELow?

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
Arguments

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

Examples

MEASUREMENT:CH2:REFLEVELS:PERCENT:RISELOW 10.0000 sets the reference levels for the rising edge to 10%.


MEASUREMENT:CH<x>:REFLEVELS:PERCENT:RISEMID

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement ref level method is set to percent. The channel number is specified by x.

Group Measurement


Arguments <NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

Examples MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEMID 50.0000 sets the MID reference level for the rising edge to 50%.

MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEMID? might return :MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEMID 50.0000 indicating the MID reference level for the rising edge is set to 50%.

MEASUREMENT:CH<x>:REFLEVELS:PERCENT:TYPE

This command specifies or queries the reference level percent type for the measurement. The channel number is specified by x.

Group Measurement

Syntax MEASUREMENT:CH<x>:REFLEVELS:PERCENT:TYPE {TENNinety|TWENtyeighty|CUSTom}
MEASUrement:CH<x>:REFLeveLs:PERCent:TYPE?

Arguments

TENNinety specifies reference levels at the 10 and 90% levels.
TWENTyeighty specifies reference levels at the 20 and 80% levels.
CUStom specifies custom reference levels.

Examples

MEASUREMENT:CH2:REFLEVELS:PERCENT:TYPE TENNINETY sets the reference levels to the 10 and 90% levels.

MEASUrement:CLOCKRecovery:ADVanced:METHod

This command sets or queries the global advanced clock recovery method. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group Measurement

Syntax MEASUrement:CLOCKRecovery:ADVanced:METHod {NONE|NOMinal|PATTern}
MEASUrement:CLOCKRecovery:ADVanced:METHod?

Arguments

NONE sets to use no advanced CRD method.
NOMinal sets the advanced CRD method to Nominal Data Rate.
PATTern sets the advanced CDR method to use a Known Data Pattern.

Examples

MEASUREMENT:CLOCKRECOVERY:ADVANCED:METHOD NOMINAL sets the CDR method to use a Known Data Pattern.
MEASUREMENT:CLOCKRECOVERY:ADVANCED:METHOD? might return :MEASUREMENT:CLOCKRECOVERY:ADVANCED:METHOD NONE indicating that no advanced CRD method will be used.
MEASUREMENT:CLOCKRecovery:CLOCKFrequency

This command sets or queries the global clock frequency used when fixed constant clock recovery is used for the measurement. This will affect measurements whose :MEASUREMENT MEAS<x>:CLOCKRecovery:GLOBAL flag is set to 1.

Group Measurement

Syntax

MEASUREMENT:CLOCKRecovery:CLOCKFrequency <NR3>
MEASUREMENT:CLOCKRecovery:CLOCKFrequency?

Arguments

<NR3> is the global clock frequency used with Constant Clock - Fixed clock recovery method.

Examples

MEASUREMENT:CLOCKRECOVERY:CLOCKFREQUENCY 2.0E+9 sets the clock frequency to 2.0 GHz.


MEASUREMENT:CLOCKRecovery:CLOCKMultiplier

This command sets or queries the global clock multiplier used when explicit clock recovery is used for the measurement. This will affect measurements whose :MEASUREMENT MEAS<x>:CLOCKRecovery:GLOBAL flag is set to 1.

Group Measurement

Syntax

MEASUREMENT:CLOCKRecovery:CLOCKMultiplier <NR3>
MEASUREMENT:CLOCKRecovery:CLOCKMultiplier?

Arguments

<NR3> is the global clock multiplier.

Examples

MEASUREMENT:CLOCKRECOVERY:CLOCKMULTIPLIER 1.000 sets the clock multiplier to 1.000.

MEASUREMENT:CLOCKRECOVERY:CLOCKMULTIPLIER? might return :MEASUREMENT:CLOCKRECOVERY:CLOCKMULTIPLIER 1.0000 indicating the clock multiplier is set to 1.0000.
MEASUrement:CLOCKRecovery:CONSTCLOCKMODE

This command sets or queries the global constant clock mode used when constant clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBAL flag is set to 1.

Group Measurement

Syntax MEASUrement:CLOCKRecovery:CONSTCLOCKMODE {MEAN|MEDian|FIXed} MEASUrement:CLOCKRecovery:CONSTCLOCKMODE?

Arguments MEAN sets the constant clock mode to MEAN.
MEDian sets the constant clock mode to MEDian.
FIXed sets the constant clock mode to FIXed.

Examples MEASUREMENT:CLOCKRECOVERY:CONSTCLOCKMODE MEAN sets the constant clock mode to MEAN.
MEASUREMENT:CLOCKRECOVERY:CONSTCLOCKMODE? might return :MEASUREMENT:CLOCKRECOVERY:CONSTCLOCKMODE MEAN indicating the constant clock mode is set to MEAN.

MEASUrement:CLOCKRecovery:DAMPing

This command sets or queries the global damping value used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBAL flag is set to 1.

Group Measurement

Syntax MEASUrement:CLOCKRecovery:DAMPing <NR3> MEASUrement:CLOCKRecovery:DAMPing?

Arguments <NR3> is the global clock recovery damping value.

Examples MEASUREMENT:CLOCKRECOVERY:DAMPING
:MEASUREMENT:CLOCKRECOVERY:DAMPING? might return :MEASUREMENT:CLOCKRECOVERY:DAMPING 700.0000E-3 indicating the damping value is set to
**MEASUrement:CLOCKRecovery:DATAPath**

This command sets or queries the global file containing the data pattern used when known data pattern clock recovery is used for the measurement. This will affect measurements whose :MEASUrement MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

**Group** Measurement

**Syntax**

```
MEASUrement:CLOCKRecovery:DATAPath <QString>
MEASUrement:CLOCKRecovery:DATAPath?
```

**Arguments**

<QString> is the path and filename, in quotes, of the file containing the data pattern.

**Examples**

```
MEASUREMENT:CLOCKRECOVERY:DATAPATH "C:/E:" sets the data path to C:/E:

```

**MEASUrement:CLOCKRecovery:DATARate**

This command sets or queries the global nominal data bit rate used when nominal data rate clock recovery is used for the measurement. This will affect measurements whose :MEASUrement MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

**Group** Measurement

**Syntax**

```
MEASUrement:CLOCKRecovery:DATARate <NR3>
MEASUrement:CLOCKRecovery:DATARate?
```

**Arguments**

<NR3> is the global value for the Nominal data rate.

**Examples**

```
MEASUREMENT:CLOCKRECOVERY:DATARATE 2.0e+9 sets the data rate for clock recovery to 2.0 GHz.

MEASUREMENT:CLOCKRECOVERY:DATARATE? might return :MEASUREMENT:CLOCKRECOVERY:DATARATE 2.5000E+9 indicating the data rate is 2.5 GHz.
```
MEASUrement:CLOCKRecovery:EXPLICITCLOCKMODE

This command sets or queries the global explicit clock mode used when explicit clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBAL flag is set to 1.

**Group**  
Measurement

**Syntax**  
MEASUrement:CLOCKRecovery:EXPLICITCLOCKMODE {EDGE|PLL}  
MEASUrement:CLOCKRecovery:EXPLICITCLOCKMODE?

**Arguments**  
EDGE sets the clock mode to clock edge.  
PLL sets the clock mode to phase locked loop.

**Examples**  
MEASUREMENT:CLOCKRECOVERY:EXPLICITCLOCKMODE EDGE sets the clock mode to edge.  
MEASUREMENT:CLOCKRECOVERY:EXPLICITCLOCKMODE? might return :MEASUREMENT:CLOCKRECOVERY:EXPLICITCLOCKMODE EDGE indicating the clock mode is edge.

MEASUrement:CLOCKRecovery:JTFBandwidth

This command sets or queries the global JTF bandwidth used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBAL flag is set to 1.

**Group**  
Measurement

**Syntax**  
MEASUrement:CLOCKRecovery:JTFBandwidth <NR3>  
MEASUrement:CLOCKRecovery:JTFBandwidth?

**Arguments**  
<NR3> is the global clock recovery JTF bandwidth.

**Examples**  
MEASUREMENT:CLOCKRECOVERY:JTFBANDWIDTH 2.0e6 sets the bandwidth to 2.0 MHz.  
MEASUREMENT:CLOCKRECOVERY:JTFBANDWIDTH? might return :MEASUREMENT:CLOCKRECOVERY:JTFBANDWIDTH 1.0000E+6 indicating the bandwidth is 1.0 MHz.
**MEASUrement:CLOCKRecovery:LOOPBandwidth**

This command sets or queries the global loop bandwidth used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUrement MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

**Group**
Measurement

**Syntax**

```
MEASUrement:CLOCKRecovery:LOOPBandwidth <NR3>
MEASUrement:CLOCKRecovery:LOOPBandwidth?
```

**Arguments**

<NR3> is the global loop bandwidth.

**Examples**

```
MEASUREMENT:CLOCKRECOVERY:LOOPBANDWIDTH 2.0e6 sets the loop bandwidth to 2.0 MHz.

MEASUREMENT:CLOCKRECOVERY:LOOPBANDWIDTH? might return :MEASUREMENT:CLOCKRECOVERY:LOOPBANDWIDTH 1.0000E+6 indicating the loop bandwidth is 1.0 MHz.
```

**MEASUrement:CLOCKRecovery:MEANAUTOCalculate**

This command sets or queries how often the clock is calculated when constant clock recovery is used for the measurement. This will affect measurements whose :MEASUrement MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

**Group**
Measurement

**Syntax**

```
MEASUrement:CLOCKRecovery:MEANAUTOCalculate {FIRST|EVERY}
MEASUrement:CLOCKRecovery:MEANAUTOCalculate?
```

**Arguments**

FIRST calculates the clock on the first acquisition.
EVERY calculates the clock on every acquisition.

**Examples**

```
MEASUREMENT:CLOCKRECOVERY:MEANAUTOCALCULATE EVERY calculates the clock on every acquisition.

MEASUREMENT:CLOCKRECOVERY:MEANAUTOCALCULATE? might return :MEASUREMENT:CLOCKRECOVERY:MEANAUTOCALCULATE FIRST indicating the clock is calculated on the first acquisition.
```
**MEASUrement:CLOCKRecovery:METHod**

This command sets or queries the global clock recovery method for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

- **Group**: Measurement
- **Syntax**: `MEASUrement:CLOCKRecovery:METHod {PLL|CONSTANTCLOCK|EXPLICITCLOCK}`
- **Arguments**: PLL specifies using the phase locked loop. CONSTANTCLOCK specifies using a constant clock. EXPLICITCLOCK specifies using an explicit clock.
- **Examples**: `MEASUREMENT:CLOCKRECOVERY:METHOD EXPLICITCLOCK` specifies using an explicit clock.
  `MEASUREMENT:CLOCKRECOVERY:METHOD?` might return :MEASUREMENT:CLOCKRECOVERY:METHOD PLL indicating the method is set to PLL.

**MEASUrement:CLOCKRecovery:MODEl**

This command sets or queries the global phase locked loop (PLL) clock recovery model used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

- **Group**: Measurement
- **Syntax**: `MEASUrement:CLOCKRecovery:MODEl {TYPE1|TYPE2}`
- **Arguments**: TYPE1 PLL clock recovery uses PLL model type I. TYPE2 PLL clock recovery uses PLL model type II.
Examples

MEASUREMENT:CLOCKRECOVERY:MODEL TYPE2 sets the model to type 2.

MEASUREMENT:CLOCKRECOVERY:MODEL? might return
:MEASUREMENT:CLOCKRECOVERY:MODEL TYPE1 indicating the
recovery model is type 1.

MEASUREMENT:CLOCKRecovery:NOMINALOFFSET

This command sets or queries the global offset value used when explicit clock
recovery is used for the measurement. This will affect measurements whose
:MEASUREMENT MEAS<x>:CLOCKRecovery:GLOBAL flag is set to 1.

Group Measurement

Syntax

MEASUREMENT:CLOCKRecovery:NOMINALOFFSET <NR3>
MEASUREMENT:CLOCKRecovery:NOMINALOFFSET?

Arguments

<NR3> is the global clock offset.

Examples

MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET 1.0 sets the offset to 1.0.

MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET? might return
:MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET 0.0E+0 indicating the
offset is set to 0.0.

MEASUREMENT:CLOCKRecovery:NOMINALOFFSET:SELECTIONTYPE

This command sets or queries the global offset type used when explicit clock
recovery is used for the measurement. This will affect measurements whose
:MEASUREMENT MEAS<x>:CLOCKRecovery:GLOBAL flag is set to 1.

Group Measurement

Syntax

MEASUREMENT:CLOCKRecovery:NOMINALOFFSET:SELECTIONTYPE
{AUTO\|MANUAL}
MEASUREMENT:CLOCKRecovery:NOMINALOFFSET:SELECTIONTYPE?

Arguments

AUTO sets the selection type to AUTO.
MANUAL sets the selection type to MANUAL.
Examples

MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET:SELECTIONTYPE AUTO sets the selection type to auto.


MEASUREMENT:CLOCKRECOVERY:STANDARD

This command sets or queries the global communications standard used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUREMENT:MEAS<x>:CLOCKRECOVERY:GLOBAL flag is set to 1.

Group Measurement

Syntax

MEASUREMENT:CLOCKRECOVERY:STANDARD {CUSTOM|ENET100|FW1394BS400B|FW1394BS800B|FW1394BS1600B|FB1|FB2|FC133|FC266|FC531|FC1063|FC2125|FC4250|FC8500|ENET1000|IBA2500|IBA_GEN2|OC1|OC3|OC12|OC48|PCIE_GEN1|PCIE_GEN2|PCIE_GEN3|RIO125|RIO250|RIO3125|SAS15_NOSSC|SAS3_NOSSC|SAS6_NOSSC|SAS12_NOSSC|SAS15_SSC|SAS3_SSC|SAS6_SSC|SAS12_SSC|SATA.Gen1|SATA.Gen2|SATA.Gen3|USB3|XAUI|XAUI Gen2}

Arguments Arguments are the global clock recovery standards.

Examples

MEASUREMENT:CLOCKRECOVERY:STANDARD PCIE_GEN3 sets the standard to PCIE_GEN3.


MEASUREMENT:DELETE (No Query Form)

The command deletes the specified measurement.

Group Measurement

Syntax MEASUREMENT:DELETE <QString>
Arguments  &lt;QString&gt; is the measurement to delete. Argument is of the form "MEAS&lt;NR1&gt;" where &lt;NR1&gt; is ≥1.

Examples  MEASUREMENT:DELETE "MEAS4" deletes measurement 4.

**MEASUrement:DIRacmodel**

This command sets or queries the dirac model used to separate random from deterministic jitter for jitter measurements.

Group  Measurement

Syntax  MEASUrement:DIRacmodel {PCIExpress|FIBREchannel}
        MEASUrement:DIRacmodel?

Arguments  PCIExpress specifies the PCIExpress dirac model.
            FIBREchannel specifies the FIBREchannel dirac model.

Examples  MEASUREMENT:DIRACMODEL FIBRECHANNEL sets the dirac model to FIBRECHANNEL.
           MEASUREMENT:DIRACMODEL? might return :MEASUREMENT:DIRACMODEL PCIEXPRESS indicating the dirac model is set to PCIECPRESS.

**MEASUrement:DISPLAYUnits**

This command sets or queries the display units used for jitter summary measurements.

Group  Measurement

Syntax  MEASUrement:DISPLAYUnits {SEConds|UNITINtervals}

Arguments  SEConds set the display units to seconds.
            UNITINtervals set the display units to unit intervals.

Examples  MEASUREMENT:DISPLAYUNITS UNITINTERVALS sets the display units to unit intervals.
MEASUREMENT:DISPLAYUNITS? might return :MEASUREMENT:DISPLAYUNITS SECONDS indicating the display units are seconds.

**MEASUrement:EDGE<x>**

This command sets or queries the type of the edge for the measurement.

**Group** Measurement

**Syntax** MEASUREMENT:EDGE<x> {RISE|FALL|BOTH}

**Arguments**
- FALL specifies the falling edge of the waveform.
- RISE specifies the rising edge of the waveform.
- BOTH specifies both a rising and falling edge of the waveform.

**Examples**
MEASUREMENT:EDGE2
MEASUREMENT:EDGE2? might return:MEASUREMENT:EDGE2? BOTH indicating

**MEASUrement:EYERENDER**

This command sets or queries the state of high-performance eye rendering for an eye diagram.

**Group** Measurement

**Syntax** MEASUREMENT:EYERENDER {<NR1>|OFF|ON}
MEASUREMENT:EYERENDER?

**Arguments**
- <NR1> = 0 disables high-performance eye rendering for an eye diagram, any other value turns this feature on.
- OFF disables high-performance eye rendering for an eye diagram.
- ON enables high-performance eye rendering for an eye diagram.
Examples

MEASUREMENT:EYERENDER
MEASUREMENT:EYERENDER? might return :MEASUREMENT:EYERENDER 1 indicating high-performance eye rendering is on.

MEASUREMENT:FILTers:BLANKingtime

This command sets or queries the global filter blanking time.

Group Measurement

Syntax

MEASUREMENT:FILTers:BLANKingtime <NR3>
MEASUREMENT:FILTers:BLANKingtime?

Arguments <NR3> is the current filter blanking time.

Examples

MEASUREMENT:FILTers:BLANKINGTIME 3.0 sets the blanking time to 3.0.
MEASUREMENT:FILTers:BLANKINGTIME? might return :MEASUREMENT:FILTers:BLANKINGTIME 4.0000 indicating the blanking time is 4.0.

MEASUREMENT:FILTers:HIGHPass:FREQ

This command sets or queries the global high pass filter frequency for the measurement.

Group Measurement

Syntax

MEASUREMENT:FILTers:HIGHPass:FREQ <NR3>
MEASUREMENT:FILTers:HIGHPass:FREQ?

Arguments <NR3> is the current high pass filter frequency.

Examples

MEASUREMENT:FILTers:HIGHPASS:FREQ 20.0e6 sets the frequency to 20 MHz.
**MEASUrement:FILTers:HIGHPass:SPEC**

This command sets or queries the global high pass filter order for the measurement.

**Group** Measurement

**Syntax**

MEASUREMENT:FILTers:HIGHPass:SPEC {NONE|FIRST|SECOND|THIRD}

MEASUREMENT:FILTers:HIGHPass:SPEC?

**Arguments**

NONE specifies no high pass filter.

FIRST specifies a first order high pass filter.

SECOND specifies a second order high pass filter.

THIRD specifies a third order high pass filter.

**Examples**

MEASUREMENT:FILTers:HIGHPass:SPEC NONE specifies no high pass filter.

MEASUREMENT:FILTers:HIGHPass:SPEC? might return

:MEASUREMENT:FILTers:HIGHPass:SPEC SECOND indicating a second order high pass filter.

**MEASUrement:FILTers:LOWPass:FREQ**

This command sets or queries the global low pass filter cutoff frequency for the measurement.

**Group** Measurement

**Syntax**

MEASUREMENT:FILTers:LOWPass:FREQ <NR3>

MEASUREMENT:FILTers:LOWPass:FREQ?

**Arguments**

<NR3> is the current low pass filter frequency.

**Examples**

MEASUREMENT:FILTers:LOWPass:FREQ 20.0e6 sets the low pass frequency to 20 MHz.

MEASUREMENT:FILTers:LOWPass:FREQ? might return

:MEASUREMENT:FILTers:LOWPass:FREQ 10.0000E+6 indicating the low pass frequency is 10.0 MHz.
MEASUrement:FILTers:LOWPass:SPEC

This command sets or queries the global low pass filter order for the measurement.

Group Measurement


Arguments NONE specifies no low pass filter.
FIRST specifies a first order low pass filter.
SECOND specifies a second order low pass filter.
THIRD specifies a third order low pass filter.

Examples MEASUREMENT:FILTERS:LOWPASS:SPEC SECOND specifies a second order low pass filter.

MEASUrement:FILTers:RAMPtime

This command sets or queries the global filter ramp time for the measurement.

Group Measurement

Syntax MEASUrement:FILTers:RAMPtime <NR3> MEASUrement:FILTers:RAMPtime?

Arguments <NR3> is the current filter ramp time.

Examples MEASUREMENT:FILTERS:RAMPTIME 3.0 sets the ramp time to 3.0.
MEASUREMENT:GATING

This command sets or queries the global gating type for the measurement.

Group            Measurement

Syntax           MEASUREMENT:GATING {NONE|SCREEN|CURSOR|LOGIC|SEARCH}
                 MEASUREMENT:GATING?

Arguments
NONE turns off measurement gating.
SCREEN turns on gating, using the left and right edges of the screen.
CURSOR limits measurements to the portion of the waveform between the vertical bar cursors, even if they are off screen.
LOGIC specifies that measurements are taken only on the portion of the waveform where the logic source is in the active state.
SEARCH specifies that measurements are taken based on search criteria.

Examples
MEASUREMENT:GATING SCREEN turns on measurement gating, using the left and right edges of the screen.
MEASUREMENT:GATING? might return :MEASUREMENT:GATING CURSOR, indicating that measurements are limited to the portion of the waveform between the vertical bar cursors.

MEASUREMENT:GATING:ACTIVE

This command sets or queries the global gating active level used for logic gating.

Group            Measurement

Syntax           MEASUREMENT:GATING:ACTIVE {HIGH|LOW}
                 MEASUREMENT:GATING:ACTIVE?

Arguments
HIGH specifies the gate is HIGH.
LOW specifies the gate is LOW.
Examples

MEASUREMENT:GATING:ACTIVE LOW specifies a low gate.

MEASUREMENT:GATING:ACTIVE? might return
:MEASUREMENT:GATING:ACTIVE HIGH indicating the gate is high.

MEASUREMENT:GATING:HYSTERESIS

This command sets or queries the global gating hysteresis value used for logic gating.

Group Measurement

Syntax

MEASUREMENT:GATING:HYSTERESIS <NR3>
MEASUREMENT:GATING:HYSTERESIS?

Arguments

<NR3> is the gating hysteresis.

Examples

MEASUREMENT:GATING:HYSTERESIS 40.0e-3 sets the hysteresis to 40 mV.

MEASUREMENT:GATING:HYSTERESIS? might return
:MEASUREMENT:GATING:HYSTERESIS 30.0000E-3 indicating the hysteresis is 30 mV.

MEASUREMENT:GATING:LOGICSource

This command sets or queries the gating data source used for logic gating.

Group Measurement

Syntax

MEASUREMENT:GATING:LOGICSource {CH<x>|MATH<x>|REF<x>}
MEASUREMENT:GATING:LOGICSource?

Arguments Arguments are the logic gating source.

Examples

MEASUREMENT:GATING:LOGICSource MATH1 sets the logic source to MATH 1.

MEASUREMENT:GATING:LOGICSource? might return
:MEASUREMENT:GATING:LOGICSource CH6 indicating the logic source is channel 6.
MEASUrement:GATing:MIDRef

This command sets or queries the global gating mid ref value used for logic gating.

Group    Measurement

Syntax   MEASUrement:GATing:MIDRef <NR3>
          MEASUrement:GATing:MIDRef?

Arguments  <NR3> is the mid ref value for gating.

Examples  MEASUREMENT:GATING:MIDREF 2.0 set the gating mid ref to 2.0 V.
          MEASUREMENT:GATING:MIDREF? might return
          :MEASUREMENT:GATING:MIDREF 1.5000 indicating the gating mid ref is 1.5 V.

MEASUrement:GATing:SEARCHSource

This command sets or queries the global gating search source when the gating type is search.

Group    Measurement

Syntax   MEASUrement:GATing:SEARCHSource SEARCH1
          MEASUrement:GATing:SEARCHSource?

Arguments  Argument is the search source.

Examples  MEASUREMENT:GATING:SEARCHSOURCE SEARCH1 set the search source to search 1.
          MEASUREMENT:GATING:SEARCHSOURCE? might return
          :MEASUREMENT:GATING:SEARCHSOURCE UNDEFINED indicating the search source is not defined.

MEASUrement:INTERp

This command sets or queries the interpolation mode used to locate edge crossings.

Group    Measurement
Commands listed in alphabetical order

Syntax

MEASUrement:INTERp {AUTO|SINX|LINear}
MEASUrement:INTERp?

Arguments

AUTO automatically selects the interpolation mode.
SINX specifies sin(x)/x interpolation, where acquired points are fit to a curve.
LINear specifies linear interpolation, where acquired points are connected with straight lines.

Examples

MEASUREMENT:INTERP SINX set the interpolation mode to sin(x)/x.
MEASUREMENT:INTERP? might return :MEASUREMENT:INTERP AUTO indicating the interpolation mode is auto.

MEASUrement:JITTermodel

This command sets or queries the model used to separate random from deterministic jitter for jitter measurements.

Group

Measurement

Syntax

MEASUrement:JITTermodel {SPECTRAL|SPECTRALBUJ}
MEASUrement:JITTermodel?

Arguments

SPECTRAL sets the jitter separation model to spectral only.
SPECTRALBUJ sets the jitter separation model to spectral plus BUJ.

Examples

MEASUREMENT:JITTERMODEL SPECTRAL set the jitter model to spectral.
MEASUREMENT:JITTERMODEL? might return :MEASUREMENT:JITTERMODEL SPECTRAL indicating the jitter model is set to spectral.

MEASUrement:LIST? (Query Only)

This query returns a comma separated list of all currently defined measurements.

Group

Measurement

Syntax

MEASUrement:LIST?
Returns

Returns a list of all currently defined measurements.

Examples

MEASUREMENT:LIST? might return :MEASUREMENT:LIST MEAS1,MEAS2,MEAS3,MEAS4,MEAS5,MEAS6 indicating 6 measurements are defined.

MEASUREMENT:LOCKRJ

This command sets or queries the state of RJ locking.

Group Measurement

Syntax

MEASUREMENT:LOCKRJ {OFF|ON|<NR1>}
MEASUREMENT:LOCKRJ?

Arguments

ON indicates that RJ locking is active.
OFF indicates that RJ locking is off.
<NR1> a 0 turns off RJ locking, any other value activates RJ locking.

Examples

MEASUREMENT:LOCKRJ ON activates RJ locking

MEASUREMENT:LOCKRJValue

This command sets or queries the RJ lock value.

Group Measurement

Syntax

MEASUREMENT:LOCKRJValue <NR3>
MEASUREMENT:LOCKRJValue?

Arguments

<NR3> default value = 1e-12; minimum value = 1e-15; maximum value = 1.

Examples

MEASUREMENT:LOCKRJVALUE 1 sets the lock value to the maximum value of 1.
MEASUREMENT:LOCKRJVALUE? might return :MEASUREMENT:LOCKRJVALUE 1.0000E-12, indicating the lock value is set to the default value of 1e-12.
**MEASUrement:MATH<x>:REFLevels:ABSolute:FALLHigh**

This command sets or queries the value used as the high reference level of the falling edge when the measurement ref level method is set to absolute. The math number is specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLHigh <NR3>  
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLHigh?

**Arguments**  
<NR3> is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHOD is set to Absolute.

**Examples**  
MEASUrement:MATH1:REFLevels:ABSolute:FALLHigh 1.5 sets the reference level to 1.5 V.  
MEASUrement:MATH2:REFLevels:ABSolute:FALLHigh? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000 indicating the reference level is 1.0 V.

**MEASUrement:MATH<x>:REFLevels:ABSolute:FALLLow**

This command sets or queries the value used as the low reference level of the falling edge when the measurement ref level method is set to absolute. The math number is specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLLow <NR3>  
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLLow?

**Arguments**  
<NR3> is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHOD is set to Absolute.

**Examples**  
MEASUrement:MATH2:REFLevels:ABSolute:FALLLow -1.5 sets the reference level to -1.5 V.  
MEASUrement:MATH2:REFLevels:ABSolute:FALLLow? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLLOW -1.0000 indicating the reference level is -1.0 V.
**MEASUrement:MATH<x>:REFLevels:ABSolute:FALLMid**

This command sets or queries the value used as the mid reference level of the falling edge when the measurement ref level method is set to absolute. The math number is specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLMid <NR3>  
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLMid?

**Arguments**  
<NR3> is the mid reference level (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Absolute.

**Examples**  
MEASUrement:MATH2:REFLevels:ABSolute:FALLMid 0.0 sets the reference level to 0.0 V.  
MEASUrement:MATH2:REFLevels:ABSolute:FALLMid? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0 indicating the reference level is 0.0 V.

**MEASUrement:MATH<x>:REFLevels:ABSolute:HYSTeresis**

This command sets or queries the value of the hysteresis of the reference level when the measurement ref level method is set to absolute. The math number is specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MATH<x>:REFLevels:ABSolute:HYSTeresis <NR3>  
MEASUrement:MATH<x>:REFLevels:ABSolute:HYSTeresis?

**Arguments**  
<NR3> is the hysteresis value used for the autoset.

**Examples**  
MEASUrement:MATH2:REFLevels:ABSolute:HYSTeresis 20.0E-3 sets the hysteresis to 20.0 mV.  
MEASUrement:MATH2:REFLevels:ABSolute:HYSTeresis? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:HYSTHERESIS 30.0000E-3 indicating the hysteresis value is 30.0 mV.
**MEASUrement:MATH<x>:REFLevels:ABSolute:RISEHigh**

This command sets or queries the value used as the high reference level of the rising edge when the measurement ref level method is set to absolute. The math number is specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEHigh <NR3>  
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEHigh?

**Arguments**  
<NR3> is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHod is set to Absolute.

**Examples**  
MEASUrement:MATH2:REFLevels:ABSolute:RISEHigh 1.5 sets the reference level to 1.5 V.  
MEASUrement:MATH2:REFLevels:ABSolute:RISEHigh? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000 indicating the reference level is 1.0 V.

**MEASUrement:MATH<x>:REFLevels:ABSolute:RISELow**

This command sets or queries the value used as the low reference level of the rising edge when the measurement ref level method is set to absolute. The math number is specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MATH<x>:REFLevels:ABSolute:RISELow <NR3>  
MEASUrement:MATH<x>:REFLevels:ABSolute:RISELow?

**Arguments**  
<NR3> is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHod is set to Absolute.

**Examples**  
MEASUrement:MATH2:REFLevels:ABSolute:RISELow -1.5 sets the reference level to -1.5 V.  
MEASUrement:MATH2:REFLevels:ABSolute:RISELow? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISELOW 1.0000 indicating the reference level is -1.0 V.
**MEASUrement:MATH<x>:REFLevels:ABSolute:RISEMid**

This command sets or queries the value used as the mid reference level of the rising edge when the measurement ref level method is set to absolute. The math number is specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEMid <NR3>  
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEMid?

**Arguments**  
<NR3> is the mid reference level (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Absolute.

**Examples**  
MEASUrement:MATH2:REFLevels:ABSolute:RISEMid 30.0E-3 sets the reference level to 30 mV.  
MEASUrement:MATH2:REFLevels:ABSolute:RISEMid? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0 indicating the reference level is 0.0 V.

**MEASUrement:MATH<x>:REFLevels:ABSolute:TYPE**

This command sets or queries the reference level type for the measurement. The math number is specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MATH<x>:REFLevels:ABSolute:TYPE {SAME|UNIQue}  
MEASUrement:MATH<x>:REFLevels:ABSolute:TYPE?

**Arguments**  
SAME specifies that the absolute levels are set the same.  
UNIQue specifies that the absolute levels can be set independently.

**Examples**  
MEASUrement:MATH2:REFLevels:ABSolute:TYPE UNIQUE specifies that the absolute levels can be set independently.  
MEASUrement:MATH2:REFLevels:ABSolute:TYPE? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:TYPE SAME indicating that the absolute levels are set the same.
MEASUrement:MATH<x>:REFLevels:BASETop

This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the measurement. The math number is specified by x.

**Group** Measurement

**Syntax**

```
MEASUrement:MATH<x>:REFLevels:BASETop
{AUTO|MINMax|MEANhistogram| MODEhistogram|EYEhistogram}
MEASUrement:MATH<x>:REFLevels:BASETop?
```

**Arguments**

- **AUTO** automatically chooses a reference level method.
- **MINMax** specifies that reference levels are relative to the measurement MIN and MAX
- **MEANhistogram** specifies that reference levels are relative to the histogram mean BASE and TOP
- **MODEhistogram** specifies that reference levels are relative to the histogram mode BASE and TOP
- **EYEhistogram** specifies that reference levels are relative to the eye histogram BASE and TOP

**Examples**

```
MEASUrement:MATH1:REFLevels:BASETop MINMax specifies that reference levels are relative to the measurement MIN and MAX

```

MEASUrement:MATH<x>:REFLevels:METHOD

This command sets or queries the method used to calculate reference levels for the measurement. The math number is specified by x.

**Group** Measurement

**Syntax**

```
MEASUrement:MATH<x>:REFLevels:METHOD {PERCent|ABSolute}
MEASUrement:MATH<x>:REFLevels:METHOD?
```
Commands listed in alphabetical order

Arguments

PERCent specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUREMENT:MATH<x>:REFlevel:PERCent commands.

ABSolute specifies that the reference levels are set explicitly using the MEASUREMENT:MATH<x>:REFLevel:ABSolute commands. This method is useful when precise values are required.

Examples

MEASUREMENT:MATH2:REFLevels:METHod ABSOLUTE specifies that the reference levels are set explicitly

MEASUREMENT:MATH2:REFLevels:METHod? might return

:MEASUREMENT:MATH2:REFLEVELS:METHOD PERCENT indicating reference levels are in percent relative to HIGH and LOW.

**MEASUREMENT:MATH<x>:REFLevels:PERCent:FALLHigh**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement ref level method is set to percent. The math number is specified by x.

**Group**

Measurement

**Syntax**

MEASUREMENT:MATH<x>:REFLevels:PERCent:FALLHigh <NR3>

MEASUREMENT:MATH<x>:REFLevels:PERCent:FALLHigh?

**Arguments**

<NR3> is the percentage (where 100% is equal to HIGH) used to calculate the high reference level when the measurement Ref level method is set to Percent.

**Examples**

MEASUREMENT:MATH2:REFLevels:PERCent:FALLHigh 95 sets the reference level to 95% of TOP.

MEASUREMENT:MATH2:REFLevels:PERCent:FALLHigh? might return

:MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLHIGH 90.0000 indicating the reference level is set to 90% of TOP.

**MEASUREMENT:MATH<x>:REFLevels:PERCent:FALLLow**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement ref level method is set to percent. The math number is specified by x.
Commands listed in alphabetical order

<table>
<thead>
<tr>
<th>Group</th>
<th>Measurement</th>
</tr>
</thead>
</table>
| Syntax      | MEASUrement:MATH<x>:REFLeve]s:PERCent:FALLLow <NR3>  
              MEASUrement:MATH<x>:REFLeve]s:PERCent:FALLLow? |
| Arguments   | <NR3> is the percentage (where 100% is equal to HIGH) used to calculate the mid reference level when the measurement’s Ref level method is set to Percent. |
| Examples    | MEASUrement:MATH2:REFLeve]s:PERCent:FALLLow 5 sets the reference level to 5% of TOP.  

MEASUrement:MATH<x>:REFLeve]s:PERCent:FALLMid

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement ref level method is set to percent. The math number is specified by x.

<table>
<thead>
<tr>
<th>Group</th>
<th>Measurement</th>
</tr>
</thead>
</table>
| Syntax      | MEASUrement:MATH<x>:REFLeve]s:PERCent:FALLMid <NR3>  
              MEASUrement:MATH<x>:REFLeve]s:PERCent:FALLMid? |
| Arguments   | <NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Percent. |
| Examples    | MEASUrement:MATH2:REFLeve]s:PERCent:FALLMid 50 sets the reference level to 50% of TOP.  
              MEASUrement:MATH2:REFLeve]s:PERCent:FALLMid? might return :MEASUREMENT:MATH2:REFLeve]s:PERCent:FALLMid 50.0000 indicating the reference level is set to 50% of TOP. |

MEASUrement:MATH<x>:REFLeve]s:PERCent:HYSTeresis

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level.
when the measurement ref level method is set to percent. The math number is specified by x

Group          Measurement

Syntax       MEASUREMENT:MATH<x>:REFLevels:PERCent:Hysteresis <NR3>
             MEASUREMENT:MATH<x>:REFLevels:PERCent:Hysteresis?

Arguments   <NR3> is the hysteresis value used for the autoset.

Examples   MEASUREMENT:MATH2:REFLevels:PERCent:Hysteresis 2.5 sets the hysteresis to 2.5% of MAX
           MEASUREMENT:MATH2:REFLevels:PERCent:Hysteresis? might return
           :MEASUREMENT:MATH2:REFLevels:PERCent:Hysteresis 5.0000
           indicating the hysteresis is set to 5% of MAX

**MEASUREMENT:MATH<x>:REFLevels:PERCent:RISEHigh**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement ref level method is set to percent. The math number is specified by x. The measurement number is specified by x.

Group          Measurement

Syntax       MEASUREMENT:MATH<x>:REFLevels:PERCent:RISEHigh <NR3>
             MEASUREMENT:MATH<x>:REFLevels:PERCent:RISEHigh?

Arguments   <NR3> is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

Examples   MEASUREMENT:MATH2:REFLevels:PERCent:RISEHigh 95 sets the reference level to 95% of TOP.
           MEASUREMENT:MATH2:REFLevels:PERCent:RISEHigh? might return
           :MEASUREMENT:MATH2:REFLevels:PERCent:RISEHigh 90.0000
           indicating the reference level is set to 90% of TOP.
MEASUrement:MATH<x>:REFLevels:PERCent:RISELow

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement ref level method is set to percent. The math number is specified by x.

Group  Measurement
Syntax  MEASUrement:MATH<x>:REFLevels:PERCent:RISELow <NR3>
       MEASUrement:MATH<x>:REFLevels:PERCent:RISELow?
Arguments  <NR3> is the percentage (where 100% is equal to TOP) used to calculate the mid reference level when the measurement Ref level method is set to Percent.
Examples  MEASUrement:MATH2:REFLevels:PERCent:RISELow 5 sets the reference level to 5% of TOP.
          MEASUrement:MATH2:REFLevels:PERCent:RISELow? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISELOW 10.0000 indicating the reference level is set to 10% of TOP.

MEASUrement:MATH<x>:REFLevels:PERCent:RISEMid

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement ref level method is set to percent. The math number is specified by x.

Group  Measurement
Syntax  MEASUrement:MATH<x>:REFLevels:PERCent:RISEMid <NR3>
       MEASUrement:MATH<x>:REFLevels:PERCent:RISEMid?
Arguments  <NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Percent.
Examples  MEASUrement:MATH2:REFLevels:PERCent:RISEMid 50 sets the reference level to 50% of TOP.
MEASUrement:MATH<x>:REFLevels:PERCent:TYPE

This command specifies or queries the reference level percent type for the measurement. The math number is specified by x.

Group Measurement

Syntax

MEASUrement:MATH<x>:REFLeveIs:PERCent:TYPE
{TENNinety|TWENtyeighty|CUSTom}
MEASUrement:MATH<x>:REFLeveIs:PERCent:TYPE?

Arguments

TENNinety sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

TWENtyeighty sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

CUSTom allows setting other reference level percents.

Examples

MEASUrement:MATH2:REFLeveIs:PERCent:TYPE TWENtyeighty sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

MEASUrement:MATH2:REFLeveIs:PERCent:TYPE? might return
:MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISEMID TENNINETY indicating the reference levels are set to 10%, 50% and 90%.

MEASUrement:MEASRange:MAX

This command sets or queries the global range maximum value.

Group Measurement

Syntax

MEASUrement:MEASRange:MAX <NR3>
MEASUrement:MEASRange:MAX?

Arguments

<NR3> is the maximum measurement range limit value.
Examples

MEASUREMENT:MEASRANGE:MAX 1.5 sets the maximum range limit to 1.5 V.
MEASUREMENT:MEASRANGE:MAX? might return
:MEASUREMENT:MEASRANGE:MAX 1.0000 indicating the maximum range limit is 1.0 V.

MEASUrement:MEASRange:MIN

This command sets or queries the global range minimum value.

Group Measurement

Syntax

MEASUREMENT:MEASRange:MIN <NR3>
MEASUREMENT:MEASRange:MIN?

Arguments

<NR3> is the minimum measurement range limit value.

Examples

MEASUREMENT:MEASRANGE:MIN 30.0E-3 sets the minimum measurement range limit to 30.0 mV.
MEASUREMENT:MEASRANGE:MIN? might return
:MEASUREMENT:MEASRANGE:MIN 0.0E+0 indicating minimum range limit is 0.0 V.

MEASUrement:MEASRange:STATE

This command sets or queries the global range state.

Group Measurement

Syntax

MEASUREMENT:MEASRange:STATE \{OFF|ON|0|1\}
MEASUREMENT:MEASRange:STATE?

Arguments

OFF specifies that the measurement results are not limited.
ON specifies that the measurement results are limited to results with values between the range minimum and maximum.
0 specifies that the measurement results are not limited.
1 specifies that the measurement results are limited to results with values between the range minimum and maximum.
Examples

MEASUREMENT:MEASRANGE:STATE 1 specifies that the measurement results are limited to results with values between the range minimum and maximum.

MEASUREMENT:MEASRANGE:STATE? might return :MEASUREMENT:MEASRANGE:STATE 0 indicating that the measurement results are not limited.

**MEASurement:MEAS<x>:BER**

This command sets or queries BER value for the measurement. Measurements are specified by x.

**Group**

Measurement

**Syntax**

MEASUREMENT:MEAS<x>:BER <NR3>
MEASUREMENT:MEAS<x>:BER?

**Arguments**

<NR3> is BER value for the measurement.

**Examples**

MEASUREMENT:MEAS2:BER 10.0 sets the BER for the measurement to 10.0.

MEASUREMENT:MEAS2:BER? might return :MEASUREMENT:MEAS2:BER 12.0000 indicating the measurement BER is 12.0.

**MEASurement:MEAS<x>:BER:TARGETBER**

This command sets or queries the target BER value for the measurement. Measurements are specified by x.

**Group**

Measurement

**Syntax**

MEASUREMENT:MEAS<x>:BER:TARGETBER <NR3>
MEASUREMENT:MEAS<x>:BER:TARGETBER?

**Arguments**

<NR3> is the target BER value.

**Examples**

MEASUREMENT:MEAS1:BER:TARGETBER 14.0 sets the target BER to 14.0.

**MEASUrement:MEAS<x>:BIN**

This command sets or queries the bin count for the measurement. Measurements are specified by x.

**Group** Measurement

**Syntax**

```
MEASUrement:MEAS<x>:BIN <NR3>
MEASUrement:MEAS<x>:BIN?
```

**Arguments**

<NR3> is the bin count.

**Examples**

```
MEASUrement:MEAS1:BIN 2 sets the bin count to 2.
MEASUrement:MEAS1:BIN? might return :MEASUREMENT:MEAS1:BIN 1 indicating the bin count is 1.
```

**MEASUrement:MEAS<x>:BITCfgmode**

This command sets or queries whether the measurement returns the mean or mode statistic result when the measurement type is bit amplitude/high/low. Measurements are specified by x.

**Group** Measurement

**Syntax**

```
MEASUrement:MEAS<x>:BITCfgmode {MEAN|MODE}
MEASUrement:MEAS<x>:BITCfgmode?
```

**Arguments**

MEAN specifies that the measurement returns results for each bit.

MODE specifies that the measurement returns the mode of its results.

**Examples**

```
MEASUrement:MEAS1:BITCfgmode MODE specifies that the measurement returns the mode of its results.
MEASUrement:MEAS1:BITCfgmode? might return :MEASUREMENT:MEAS1:BITCFGMODE MEAN indicating the configuration mode is set to MEAN.
```
MEASurement:MEAS<x>:BITEnd

This command sets or queries the bit end as a percentage of the unit interval. Measurements are specified by x.

Group Measurement

Syntax MEASurement:MEAS<x>:BITEnd <NR3>
MEASurement:MEAS<x>:BITEnd?

Arguments <NR3> is the bit end.

Examples MEASurement:MEAS2:BITEnd 60 sets the bit end to 60% of the unit interval. MEASurement:MEAS2:BITEnd? might return :MEASUREMENT:MEAS2:BITEND 50.0000 indicating the bit end is 50%.

MEASurement:MEAS<x>:BITPcnt

This command sets or queries the bit center as a percentage of the unit interval. Measurements are specified by x.

Group Measurement

Syntax MEASurement:MEAS<x>:BITPcnt <NR3>
MEASurement:MEAS<x>:BITPcnt?

Arguments <NR3> is the bit center percentage value to be measured for the bit type selected.

Examples MEASurement:MEAS2:BITPcnt 55 sets the bit center to 55% of the unit interval. MEASurement:MEAS2:BITPcnt? might return :MEASUREMENT:MEAS2:BITPCNT 50.0000 indicating bit center is 50% of the unit interval.

MEASurement:MEAS<x>:BITSTart

This command sets or queries the bit start as a percentage of the unit interval. Measurements are specified by x.
### MEASurement:MEAS<x>:BITStart

**Group** Measurement

**Syntax**

```
MEASUREMENT:MEAS<x>:BITSTart <NR3>
MEASUREMENT:MEAS<x>:BITSTart?
```

**Arguments**
<br>

<NR3> is the bit start.

**Examples**

```
MEASUREMENT:MEAS2:BITSTart 45 sets the bit start to 45% of the unit interval.
```

```
MEASUREMENT:MEAS2:BITSTart? might return
:MEASUREMENT:MEAS2:BITSTart 50.0000 indicating the bit start is 50% of the unit interval.
```

### MEASurement:MEAS<x>:BITType

This command sets or queries the bit type for the measurement. Measurements are specified by x.

**Group** Measurement

**Syntax**

```
MEASUREMENT:MEAS<x>:BITType {ALLBits|TRANSition|NONTRANSition}
MEASUREMENT:MEAS<x>:BITType?
```

**Arguments**
<br>

ALLBits specifies that the measurement returns results for all bits.

TRANSition specifies that the measurement returns results for transitions bit only.

NONTRANSition specifies that the measurement returns results for non-transition bits only.

**Examples**

```
MEASUREMENT:MEAS1:BITType TRANSITION specifies that the measurement returns results for transition bits only.
```

```
MEASUREMENT:MEAS1:BITType? might return
:MEASUREMENT:MEAS1:BITType ALLBITS indicating that measurements return results for all bits.
```

### MEASurement:MEAS<x>:BURSTEDGTYPE

This command sets or queries the burst edge type for the measurement. Measurements are specified by x.

**Group** Measurement

**Syntax**

```
MEASUREMENT:MEAS<x>:BURSTEDGTYPE
```

**Examples**

```
MEASUREMENT:MEAS2:BURSTEDGTYPE
```

---

2-480  MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
Commands listed in alphabetical order

Group Measurement

Syntax MEASUREMENT:MEAS<x>:BURSTEDGTYPe {RISE|FALL}
MEASUREMENT:MEAS<x>:BURSTEDGTYPe?

Arguments RISE specifies a burst with a rising edge.
FALL specifies a burst with a falling edge.

Examples MEASUREMENT:MEAS2:BURSTEDGTYPe FALL specifies a burst with a falling edge.

MEASUREMENT:MEAS<x>:CCRESULTS:ALLAcqs:MAXimum? (Query Only)

This query-only command returns the maximum cycle-cycle value for the specified measurement for all acquisitions. Measurements are specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:CCRESULTS:ALLAcqs:MAXimum?

Returns The maximum cycle-cycle statistic value for the specified measurement for all acquisitions.


MEASUREMENT:MEAS<x>:CCRESULTS:ALLAcqs:MEAN? (Query Only)

This query-only command returns the mean cycle-cycle value for the specified measurement for all acquisitions. Measurements are specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:CCRESULTS:ALLAcqs:MEAN?
Commands listed in alphabetical order

<table>
<thead>
<tr>
<th>Group</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td><code>MEAS&lt;Measurement&gt;:CCRESULTS:ALLAcqs:MEAN?</code></td>
</tr>
<tr>
<td>Returns</td>
<td>The mean cycle-cycle statistic value for the specified measurement for all acquisitions.</td>
</tr>
</tbody>
</table>

**MEASurement:MEAS<x>:CCRESUlts:ALLAcqs:MINimum? (Query Only)**

This query-only command returns the minimum cycle-cycle value for the specified measurement for all acquisitions. Measurements are specified by x.

<table>
<thead>
<tr>
<th>Group</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td><code>MEAS&lt;Measurement&gt;:CCRESULTS:ALLAcqs:MINimum?</code></td>
</tr>
<tr>
<td>Returns</td>
<td>The minimum cycle-cycle statistic value for the specified measurement for all acquisitions.</td>
</tr>
</tbody>
</table>

**MEASurement:MEAS<x>:CCRESUlts:ALLAcqs:PK2PK? (Query Only)**

This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for all acquisitions. Measurements are specified by x.

<table>
<thead>
<tr>
<th>Group</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td><code>MEAS&lt;Measurement&gt;:CCRESULTS:ALLAcqs:PK2PK?</code></td>
</tr>
<tr>
<td>Returns</td>
<td>The peak to peak cycle-cycle statistic value for the specified measurement for all acquisitions.</td>
</tr>
</tbody>
</table>
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:POPUlation? (Query Only)

This query-only command returns the population of all cycle-cycle statistics for the specified measurement for all acquisitions accumulated since statistics were last reset. Measurements are specified by x.

Group Measurement
Syntax MEASUrement:MEAS<x>:CCRESUIts:ALLAcqs:POPUlation?
Returns The population of all cycle-cycle statistics for the specified measurement accumulated over all acquisitions since statistics were last reset.

MEASUrement:MEAS<x>:CCRESUIts:ALLAcqs:STDDev? (Query Only)

This query-only command returns the standard deviation cycle-cycle for the specified measurement for all acquisitions. Measurements are specified by x.

Group Measurement
Syntax MEASUrement:MEAS<x>:CCRESUIts:ALLAcqs:STDDev?
Returns The standard deviation cycle-cycle statistic value for the specified measurement all acquisitions.

MEASUrement:MEAS<x>:CCRESUIts:CURREntacq:MAXimum? (Query Only)

This query-only command returns the maximum cycle-cycle value for the specified measurement for the current acquisition. Measurements are specified by x.

Group Measurement
Syntax: \texttt{MEASurement:MEAS<x>:CCRESUlt<ts:CURRentacq:M}AXimum?\后面的命令

Returns: The maximum cycle-cycle statistic value for the specified measurement for the current acquisition.


\textbf{MEASurement:MEAS<x>:CCRESUlt<ts:CURRentacq:MEAN? (Query Only)}

This query-only command returns the mean cycle-cycle value for the specified measurement for the current acquisition. Measurements are specified by x.

Group: Measurement

Syntax: \texttt{MEASurement:MEAS<x>:CCRESUlt<ts:CURRentacq:MEAN?}

Returns: The mean cycle-cycle statistic value for the specified measurement for the current acquisition.


\textbf{MEASurement:MEAS<x>:CCRESUlt<ts:CURRentacq:MINimum? (Query Only)}

This query-only command returns the minimum cycle-cycle value for the specified measurement for the current acquisition. Measurements are specified by x.

Group: Measurement

Syntax: \texttt{MEASurement:MEAS<x>:CCRESUlt<ts:CURRentacq:MINimum?}

Returns: The minimum cycle-cycle statistic value for the specified measurement for the current acquisition.

MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:PK2PK? (Query Only)

This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for the current acquisition. Measurements are specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:PK2PK?

Returns The peak to peak cycle-cycle statistic value for the specified measurement for the current acquisition.


MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:POPUlation? (Query Only)

This query-only command returns the population of the cycle-cycle statistics for the specified measurement for the current acquisition. Measurements are specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:POPUlation?

Returns The population of the cycle-cycle statistics for the specified measurement for the current acquisition.


MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:STDDev? (Query Only)

This query-only command returns the standard deviation cycle-cycle for the specified measurement for the current acquisition. Measurements are specified by x.
MEASUrement:MEAS<x>:CCRESUlt:CCRecovery:ADVanced:METHod

This command sets or queries the advanced clock recovery method when advanced clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement
Syntax MEASUrement:MEAS<x>:CCRecovery:ADVanced:METHod
{NONE|Nominal|PATTern}
MEASUrement:MEAS<x>:CCRecovery:ADVanced:METHod?
Arguments
Nominal sets the advanced CRD method to Nominal Data Rate.
PATTern sets the advanced CDR method to use a Known Data Pattern.
NONE sets no advanced CRD method.
Examples
MEASUrement:MEAS1:CCRecovery:ADVanced:METHod PATTERN sets the advanced CDR method to use a Known Data Pattern.

MEASUrement:MEAS<x>:CCRecovery:CLOCKFrequency

This command sets or queries the clock frequency used when fixed constant clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement
Syntax
MEASurement:MEAS<x>:CLOCKRecovery:CLOCKFrequency <NR3>
MEASurement:MEAS<x>:CLOCKRecovery:CLOCKFrequency?

Arguments
<NR3> is the clock frequency.

Examples
MEASurement:MEAS1:CLOCKRecovery:CLOCKFrequency 2.0E+9 sets the frequency to 2.0 GHz.

MEASurement:MEAS<x>:CLOCKRecovery:CLOCKMultiplier
This command sets or queries the clock multiplier used when explicit clock recovery is used for the measurement. Measurements are specified by x.

Group
Measurement

Syntax
MEASurement:MEAS<x>:CLOCKRecovery:CLOCKMultiplier <NR3>
MEASurement:MEAS<x>:CLOCKRecovery:CLOCKMultiplier?

Arguments
<NR3> is the clock multiplier.

Examples
MEASurement:MEAS1:CLOCKRecovery:CLOCKMultiplier 1.5 sets the clock multiplier to 1.5.

MEASurement:MEAS<x>:CLOCKRecovery:CONSTCLOCKMODE
This command sets or queries the constant clock mode used when constant clock recovery is used for the measurement. The measurement number is specified by x.

Group
Measurement

Syntax
MEASurement:MEAS<x>:CLOCKRecovery:CONSTCLOCKMODE {MEAN|MEDian|FIXed}
Commands listed in alphabetical order

Arguments

MEAN specifies that clock recovery uses the mean of the clock signal as the clock frequency.

MEDian specifies that clock recovery uses the mode of the clock signal as the clock frequency.

FIXed specifies that clock recovery uses the value set by the user as the clock frequency.

Examples

MEASurement:MEAS1:CLOCKRecovery:CONSTCLOCKMODE FIXED specifies that clock recovery uses the value set by the user as the clock frequency.

MEASurement:MEAS1:CLOCKRecovery:CONSTCLOCKMODE? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:CONSTCLOCKMODE MEAN indicating that clock recovery uses the mode of the clock signal as the clock frequency.

**MEASUrement:MEAS<x>:CLOCKRecovery:DAMPing**

This command sets or queries the damping value used when PLL clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement

Syntax

MEASurement:MEAS<x>:CLOCKRecovery:DAMPing <NR3>

MEASurement:MEAS<x>:CLOCKRecovery:DAMPing?

Arguments <NR3> is the clock recovery damping value.

Examples

MEASurement:MEAS1:CLOCKRecovery:DAMPing 700.0E-3 sets the damping value to 0.70.

MEASurement:MEAS1:CLOCKRecovery:DAMPing? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:DAMPING 700.0000E-3 indicating the damping value is 0.70.

**MEASUrement:MEAS<x>:CLOCKRecovery:DATAPath**

This command sets or queries the file containing the data pattern used when known data pattern clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement
Commands listed in alphabetical order

**MEASUrement:MEAS<x>:CLOCKRecovery:DATAPath**

Syntax

MEASUrement:MEAS<x>:CLOCKRecovery:DATAPath <QString>

MEASUrement:MEAS<x>:CLOCKRecovery:DATAPath?

Arguments

<QString> is the file containing the data pattern to be used for known data pattern clock recovery.

Examples

MEASUrement:MEAS1:CLOCKRecovery:DATAPath "TEST125.txt" specifies the file containing the data pattern is TEST125.txt.


**MEASUrement:MEAS<x>:CLOCKRecovery:DATARate**

This command sets or queries the nominal data bit rate when nominal data rate clock recovery is used for the measurement. Measurements are specified by x.

Group  
Measurement

Syntax

MEASUrement:MEAS<x>:CLOCKRecovery:DATARate <NR3>

MEASUrement:MEAS<x>:CLOCKRecovery:DATARate?

Arguments

<NR3> is the value for the Nominal data rate.

Examples

MEASUrement:MEAS1:CLOCKRecovery:DATARate 2.0000E+9 sets the data rate to 2.0 Gb/s.

MEASUrement:MEAS1:CLOCKRecovery:DATARate? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:DATARATE 2.5000E+9 indicating the data rate is 2.5 Gb/s.

**MEASUrement:MEAS<x>:CLOCKRecovery:EXPLICITCLOCKMODE**

This command sets or queries the explicit clock mode used when explicit clock recovery is used for the measurement. The measurement number is specified by x.

Group  
Measurement

Syntax

MEASUrement:MEAS<x>:CLOCKRecovery:EXPLICITCLOCKMODE {EDGE|PLL}
MEASurement:MEAS<x>:CLOCKRecovery:EXPLICITCLOCKMODe?

Arguments
- **EDGE** specifies the clock edge.
- **PLL** specifies the phase locked loop.

Examples
- MEASurement:MEAS1:CLOCKRecovery:EXPLICITCLOCKMODe EDGE sets the clock mode to edge.

MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal

This command sets or queries the clock recovery settings global flag for the measurement. Measurements are specified by x.

Group
- Measurement

Syntax
- MEASurement:MEAS<x>:CLOCKRecovery:GLOBal {OFF|ON|0|1}
- MEASurement:MEAS<x>:CLOCKRecovery:GLOBal?

Arguments
- **OFF** clock recovery settings are changed independently for each individual measurement.
- **ON** applies global clock recovery settings to all the measurements' clock recovery settings.
- **0** clock recovery settings are changed independently for each individual measurement.
- **1** applies global clock recovery settings to all the measurements' clock recovery settings.

Examples
- MEASurement:MEAS1:CLOCKRecovery:GLOBal 1 applies global clock recovery settings to all the measurements' clock recovery settings.
- MEASurement:MEAS1:CLOCKRecovery:GLOBal? might return :MEASUREMENT:MEAS1:CLOCKRecovery:GLOBal 0 indicating clock recovery settings are changed independently for each individual measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:JTFBandwidth

This command sets or queries the JTF bandwidth used when PLL clock recovery is used for the measurement. Measurements are specified by x.

Group  Measurement

Syntax  MEASUrement:MEAS<x>:CLOCKRecovery:JTFBandwidth <NR3>
        MEASUrement:MEAS<x>:CLOCKRecovery:JTFBandwidth?

Arguments  <NR3> is the clock recovery JTF bandwidth

Examples  MEASUrement:MEAS1:CLOCKRecovery:JTFBandwidth 1.10E+6 sets the bandwidth to 1.1 MHz.

MEASUrement:MEAS<x>:CLOCKRecovery:LOOPBandwidth

This command sets or queries the loop bandwidth used when PLL clock recovery is used for the measurement. Measurements are specified by x.

Group  Measurement

Syntax  MEASUrement:MEAS<x>:CLOCKRecovery:LOOPBandwidth <NR3>
        MEASUrement:MEAS<x>:CLOCKRecovery:LOOPBandwidth?

Arguments  <NR3> is the clock recovery loop bandwidth

Examples  MEASUrement:MEAS1:CLOCKRecovery:LOOPBandwidth 1.10E+6 sets the bandwidth to 1.1 MHz.
**MEASUrement:MEAS<x>:CLOCKRecovery:MEANAUTOCalculate**

This command sets or queries how often the clock is calculated when constant clock recovery is used for the measurement. The measurement number is specified by x.

**Group**
Measurement

**Syntax**
MEASUrement:MEAS<x>:CLOCKRecovery:MEANAUTOCalculate
{FIRST|EVERY}
MEASUrement:MEAS<x>:CLOCKRecovery:MEANAUTOCalculate?

**Arguments**
FIRST calculates the clock on the first acquisition.
EVERY calculates the clock on every acquisition.

**Examples**
MEASUrement:MEAS1:CLOCKRecovery:MEANAUTOCalculate EVERY causes the clock to be calculated on every acquisition.

**MEASUrement:MEAS<x>:CLOCKRecovery:METHod**

This command sets or queries the clock recovery method for the measurement. Measurements are specified by x.

**Group**
Measurement

**Syntax**
MEASUrement:MEAS<x>:CLOCKRecovery:METHod
{PLL|CONSTANTCLOCK|EXPLICITCLOCK}
MEASUrement:MEAS<x>:CLOCKRecovery:METHod?

**Arguments**
PLL specifies a phase locked loop clock recovery method.
CONSTANTCLOCK specifies using a constant clock.
EXPLICITCLOCK specifies using an explicit clock.

**Examples**
MEASUrement:MEAS1:CLOCKRecovery:METHod PLL specifies a phase locked loop clock recovery method.
MEASUrement:MEAS<x>:CLOCKRecovery:MODel

This command sets or queries the PLL clock recovery model used when PLL clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:CLOCKRecovery:MODel {TYPE1|TYPE2} MEASUrement:MEAS<x>:CLOCKRecovery:MODel?

Arguments Arguments are the clock recovery model type.

Examples MEASUrement:MEAS1:CLOCKRecovery:MODel TYPE1 set the clock recovery model to type 1.


MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset

This command sets or queries the offset value used when explicit clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset <NR3> MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset?

Arguments <NR3> is the clock offset.

Examples MEASUrement:MEAS1:CLOCKRecovery:NOMINALOFFset 1.0e-9 sets the clock offset to 1 ns.
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFSET:SELECTIONtype

This command sets or queries the offset type used when explicit clock recovery is used for the measurement. The measurement number is specified by x.

**Group**
Measurement

**Syntax**
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFSET:SELECTIONtype {AUTO|MANUAL}
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFSET:SELECTIONtype?

**Arguments**
AUTO automatically calculates the offset.
MANUAL allows the user to set the offset.

**Examples**
MEASUrement:MEAS1:CLOCKRecovery:NOMINALOFFSET:SELECTIONtype AUTO sets the offset selection type to auto.

MEASUrement:MEAS1:CLOCKRecovery:NOMINALOFFSET:SELECTIONtype?
might return
:MEASUREMENT:MEAS1:CLOCKRECOVERY:NOMINALOFFSET:SELECTIONTYPE MANUAL indicating the selection type is manual.

MEASUrement:MEAS<x>:CLOCKRecovery:STAndard

This command sets or queries the communications standard when PLL clock recovery is used for the measurement. The measurement number is specified by x.

**Group**
Measurement

**Syntax**
MEASUrement:MEAS<x>:CLOCKRecovery:STAndard {CUSTom|ENET100|FW1394BS400B|FW1394BS800B|FW1394BS1600B|FBD1|FBD2|FBD3|FC133|FC266|FC531|FC1063|FC2125|FC4250|FC8500|ENET1000|IBA2500|IBA_GEN2|OC1|OC3|OC12|OC48|PCIE_GEN1|PCIE_GEN2|PCIE_GEN3|RIO125|RIO250|RIO3125|SAS15_NOSSC|SAS3_NOSSC|SAS6_NOSSC|SAS12_NOSSC|SAS15_SSC|SAS3_SSC|SAS6_SSC|SAS12_SSC|SATA_GEN1|SATA_GEN2|SATA Gen3|USB3|XAUI|XAUI_GEN2}
Arguments
Arguments are the clock recovery standards.

Examples
MEASUrement:MEAS1:CLOCKRecovery:STAndard PCIE_GEN2 sets the standard to PCIE_GEN2.

MEASUrement:MEAS<x>:COMMONMode:FILTers:STATE
This command sets or queries whether a filter is used for the measurement when the measurement type is AC common mode. Measurements are specified by x.

Group
Measurement

Syntax
MEASUrement:MEAS<x>:COMMONMode:FILTers:STATE {OFF|ON|0|1}
MEASUrement:MEAS<x>:COMMONMode:FILTers:STATE?

Arguments
OFF turns the filter off.
ON turns the filter on.
0 turns the filter off.
1 turns the filter on.

Examples
MEASUrement:MEAS1:COMMONMode:FILTers:STATE ON sets the filter to on.

MEASUrement:MEAS<x>:COMMONMode:SOURCEs
This command sets or queries the number of sources for the measurement when the measurement type is AC common mode. Measurements are specified by x.

Group
Measurement

Syntax
MEASUrement:MEAS<x>:COMMONMode:SOURCEs {SINGLE|DOUBLE}
MEASUrement:MEAS<x>:COMMONMode:SOURCEs?
Arguments

SINGLE specifies a single source.
DOUBLE specifies double sources.

Examples

MEASurement:MEAS1:COMMONMode:SOURCEs SINGLE specifies a single source.

MEASUrement:MEAS<x>:CYCLeMode

This command sets or queries the cycle mode for the measurement. Measurements are specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:CYCLeMode {RECORD|CYCLE}
MEASUrement:MEAS<x>:CYCLeMode?

Arguments

RECORD specifies that the measurement is taken over the whole record.
CYCLE specifies that measurements are taken on each cycle of the source.

Examples

MEASUrement:MEAS1:CYCLeMode CYCLE sets the measurement to be taken over each cycle of the source.
MEASUrement:MEAS1:CYCLeMode? might return :MEASUREMENT:MEAS1:CYCLeMODE RECORD indicating the measurement is taken over the whole record.

MEASUrement:MEAS<x>:DELay:EDGE<x>

This command sets or queries the 'to edge' type when EDGE<x> is EDGE1 and the 'from edge' type when EDGE<x> is EDG2, for the measurement when the measurement type is DELAY. Measurements are specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:DELay:EDGE<x> {FALL|RISe|BOTH|SAMEas|OPPositeas}
MEASurement:MEAS<x>:DELAY:EDGE<x>?

Related Commands

MEASUrement:MEAS<x>:TOEdge

Arguments

FALL specifies the falling edge of the waveform.
RISE specifies the rising edge of the waveform.
BOTH specifies both a rising and falling edge of the waveform.
SAMEas specifies that both edges of the waveform are the same.
OPPositeas specifies that the edges of the waveform are not the same.

Examples

MEASUREMENT:MEAS3:DELAY:EDGE2 RISE specifies that the rising edge of the waveform be used for measurement 3.
MEASUREMENT:MEAS1:DELAY:EDGE2? might return :MEASUREMENT:MEAS1:DELAY:EDGE2 FALL, indicating that the falling edge of the waveform is being used for measurement 1.

MEASUrement:MEAS<x>:DISPlaystat:ENABle

This command turns on and off display of statistics in measurement badges in the user interface. This command affects only the display of statistics, basic statistics are computed regardless of the state of this command. Measurements are specified by x.

Group Measurement

Syntax

MEASUrement:MEAS<x>:DISPlaystat:ENABle {OFF|ON|<NR1>}
MEASUrement:MEAS<x>:DISPlaystat:ENABle?

Arguments

OFF turns off the display of statistics in measurement badges.
ON turns on the display of statistics in measurement badges.
<NR1> = 0 turns off the display of statistics in the measurement badge, any other value turns on the display of statistics.

Examples

MEASUREMENT:MEAS<x>:DISPlaystat:ENABle 0 turns off the display of statistics in the measurement badge.
Commands listed in alphabetical order

MEASUrement:MEAS<x>:DISPLAYstat:ENABLe?

This command sets or queries the edge increment value for the measurement. Measurements are specified by x.

Group: Measurement

Syntax: MEASUrement:MEAS<x>:EDGEIncre <NR3>
        MEASUrement:MEAS<x>:EDGEIncre?

Arguments: <NR3> is the measurements edge increment value.

Examples: :MEASUREMENT:MEAS1:EDGEINCRE 1 indicating the edge increment value is set to 1.0000.

MEASUrement:MEAS<x>:EDGES:FROMLevel

This command sets or queries the 'from level' edge for the measurement. Measurements are specified by x.

Group: Measurement

Syntax: MEASUrement:MEAS<x>:EDGES:FROMLevel {MID|LOW|HIGH}
        MEASUrement:MEAS<x>:EDGES:FROMLevel?

Arguments: MID specifies the MID level.
            HIGH specifies the HIGH level.
            LOW specifies the LOW level.

Examples: MEASUrement:MEAS1:EDGES:FROMLevel HIGH set the from level to the high level
MEASUrement:MEAS<x>:EDGES:LEVEL

This sets or queries the level type for the 'time outside level' measurement. Measurements are specified by x.

Group: Measurement

Syntax:

MEASUrement:MEAS<x>:EDGES:LEVEL {HIGH|LOW|BOTH} 
MEASUrement:MEAS<x>:EDGES:LEVEL?

Arguments:

HIGH specifies the HIGH level.
LOW specifies the LOW level.
BOTH specifies both the HIGH and LOW level.

Examples:

MEASUrement:MEAS1:EDGES:LEVEL LOW sets the level to the low level.
MEASUrement:MEAS1:EDGES:LEVEL? might return :MEASUREMENT:MEAS1:EDGES:LEVEL HIGH indicating the level is set to the high level.

MEASUrement:MEAS<x>:EDGES:LOWERFREQuency

This command sets or queries the lower frequency for the measurement when the measurement type is phase noise. Lower frequencies are ignored. Measurements are specified by x.

Group: Measurement

Syntax:

MEASUrement:MEAS<x>:EDGES:LOWERFREQuency <NR3> 
MEASUrement:MEAS<x>:EDGES:LOWERFREQuency?

Arguments:

<NR3> is the lower frequency of the edge.

Examples:

MEASUrement:MEAS1:EDGES:LOWERFREQuency 1.0e3 sets the lower frequency to 1 kHz.
MEASUrement:MEAS1:EDGES:LOWERFREQuency? might return :MEASUREMENT:MEAS1:EDGES:LOWERFREQUENCY 0.0E+0 indicating the lower frequency is set to 0.0 Hertz.

MEASUrement:MEAS<x>:EDGES:N

The command sets or queries the number of accumulation cycles for the measurement when the measurement type is nperiod. Measurements are specified by x.

Group Measurement

Syntax

MEASUrement:MEAS<x>:EDGES:N <NR3>
MEASUrement:MEAS<x>:EDGES:N?

Arguments

<NR3> is the maximum number of edges used by the measurement.

Examples

measurement:meas1:edges:n 2 sets the number of edges to 2.

MEASUrement:MEAS<x>:EDGES:SLEWRATEMethod

This command sets or queries the slew rate method for the measurement. Measurements are specified by x.

Group Measurement

Syntax

MEASUrement:MEAS<x>:EDGES:SLEWRATEMethod {NOMinal|DDR}
MEASUrement:MEAS<x>:EDGES:SLEWRATEMethod?

Arguments

NOMinal specifies the nominal slew rate method.
DDR specifies the DDR slew rate method.

Examples

measurement:meas1:edges:slewratemethod NOMINAL specifies the nominal slew rate method.
Commands listed in alphabetical order

measurement:meas1:edges:slewratemethod? might return
:MEASUREMENT:MEAS1:EDGES:SLEWRATEMETHOD NOMINAL indicating the
slew rate method is set to NOMINAL.

MEASUrement:MEAS<x>:EDGES:TOLevel

This command sets or queries the 'to level' edge for the measurement.
Measurements are specified by x.

Group       Measurement
Syntax      MEASUrement:MEAS<x>:EDGES:TOLevel {HIGH|MID|LOW}
            MEASUrement:MEAS<x>:EDGES:TOLevel?

Arguments   HIGH specifies the HIGH level.
            MID specifies the MID level.
            LOW specifies the LOW level.

Examples    MEASUrement:MEAS1:EDGES:TOLevel HIGH specifies the High to level.
            MEASUrement:MEAS1:EDGES:TOLevel? might return
            :MEASUREMENT:MEAS1:EDGES:TOLeVEL LOW indicating the to level
            edge is set to the Low level.

MEASUrement:MEAS<x>:EDGES:UPPERFREQuency

This command sets or queries the upper frequency for the measurement when the
measurement type is phase noise. Higher frequencies are ignored. Measurements
are specified by x.

Group       Measurement
Syntax      MEASUrement:MEAS<x>:EDGES:UPPERFREQuency <NR3>
            MEASUrement:MEAS<x>:EDGES:UPPERFREQuency?

Arguments   <NR3> is the upper frequency of the edge.

Examples    MEASUrement:MEAS1:EDGES:UPPERFREQuency 1.5 sets the upper frequency
to 1.5 MHz.
MEASUrement:MEAS1:EDGES:UPPERFREQuency? might return 
:MEASUREMENT:MEAS1:EDGES:UPPERFREQUENCY 1.0000E+6 indicating the 
upper frequency is 1.0 MHz.

**MEASUrement:MEAS<x>:EDGE<x>**

This command sets or queries the type of the specified edge, rise or fall, for the 
measurement. The measurement number is specified by x.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:EDGE<x> {RISE|FALL|BOTH}

**Arguments** RISE specifies the rising edge.
FALL specifies the falling edge.
BOTH specifies either the rising or falling edge.

**Examples** MEASUREMENT:MEAS2:EDGE1? might return :MEASUREMENT:MEAS2:EDGE1 
RISE indicating that edge 1 of measurement 2 is the rising edge.

**MEASUrement:MEAS<x>:FILTers:BLANKingtime**

This command sets or queries the filter blanking time for the measurement. 
Measurements are specified by x.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:FILTers:BLANKingtime <NR3> 
MEASUrement:MEAS<x>:FILTers:BLANKingtime?

**Arguments** <NR3> is the current filter blanking time.

**Examples** MEASUREMENT:MEAS1:FILTers:BLANKingtime 3.5 sets the blanking time to 
3.5.

MEASUREMENT:MEAS1:FILTers:BLANKingtime? might return 
:MEASUREMENT:MEAS1:FILTers:BLANKINGTIME 4.0000 indicating the filter 
blanking time is 4.0.
MEASUrement:MEAS<x>:FILTers:GLOBal

This command sets or queries the global flag for filter settings for the measurement. Measurements are specified by x.

**Group**  Measurement

**Syntax**  
MEASUrement:MEAS<x>:FILTers:GLOBal {OFF|ON|0|1}  
MEASUrement:MEAS<x>:FILTers:GLOBal?

**Arguments**  
OFF causes filter settings to be changed independently for each individual measurement.

ON applies global filter settings to all the measurements' filter settings.

0 causes filter settings to be changed independently for each individual measurement.

1 applies global filter settings to all the measurements' filter settings.

**Examples**  
MEASUrement:MEAS<x>:FILTers:GLOBal OFF causes filter settings to be changed independently for each individual measurement.


MEASUrement:MEAS<x>:FILTers:HIGHPass:FREQ

This command sets or queries the high pass filter frequency for the measurement. Measurements are specified by x.

**Group**  Measurement

**Syntax**  
MEASUrement:MEAS<x>:FILTers:HIGHPass:FREQ <NR3>  
MEASUrement:MEAS<x>:FILTers:HIGHPass:FREQ?

**Arguments**  
<NR3> is the current high pass filter frequency.

**Examples**  
MEASUrement:MEAS1:FILTers:HIGHPass:FREQ 15.0E+6 sets the high pass frequency to 15.0 MHz.
Commands listed in alphabetical order


**MEASUREMENT:MEAS<x>:FILTers:HIGHPass:SPEC**

This command sets or queries the high pass filter order for the measurement. Measurements are specified by x.

- **Group**: Measurement
- **Syntax**: MEASUREMENT:MEAS<x>:FILTers:HIGHPass:SPEC {NONE|FIRST|SECOND|THIRD}
  - MEASUREMENT:MEAS<x>:FILTers:HIGHPass:SPEC?
- **Arguments**:
  - NONE specifies no filter.
  - FIRST specifies a first-order filter.
  - SECOND specifies a second-order filter.
  - THIRD specifies a third-order filter.

**Examples**


**MEASUREMENT:MEAS<x>:FILTers:LOWPass:FREQ**

This command sets or queries the low pass filter cutoff frequency for the measurement. Measurements are specified by x.

- **Group**: Measurement
- **Syntax**: MEASUREMENT:MEAS<x>:FILTers:LOWPass:FREQ <NR3>
  - MEASUREMENT:MEAS<x>:FILTers:LOWPass:FREQ?
- **Arguments**: <NR3> is the current low pass filter frequency.
Examples

MEASurement:MEAS1:FILTers:LOWPass:FREQ 5.0E+6 sets the low pass frequency to 5.0 MHz.


MEASUrement:MEAS<x>:FILTers:LOWPass:SPEC

This command sets or queries the low pass filter order for the measurement. Measurements are specified by x.

Group
Measurement

Syntax
MEASUrement:MEAS<x>:FILTers:LOWPass:SPEC
{NONE|FIRST|SECOND|THIRD}
MEASUrement:MEAS<x>:FILTers:LOWPass:SPEC?

Arguments
NONE specifies no filter.
FIRST specifies a first-order filter.
SECOND specifies a second-order filter.
THIRD specifies a third-order filter.

Examples


MEASUrement:MEAS<x>:FILTers:RAMPtime

This command sets or queries the filter ramp time for the measurement. Measurements are specified by x.

Group
Measurement

Syntax
MEASUrement:MEAS<x>:FILTers:RAMPtime <NR3>
MEASUrement:MEAS<x>:FILTers:RAMPtime?
Arguments  

<NR3> is the current filter ramp time.

Examples  

MEASurement:MEAS1:FILTERs:RAMPtime 1.5 sets the ramp time to 1.5.

MEASurement:MEAS1:FILTERs:RAMPtime? might return
:MEASUREMENT:MEAS1:FILTERs:RAMPTIME 2.0000 indicating the ramp
time is 2.0.

**MEASUrement:MEAS<x>:FROMedge**

This command sets or queries the from edge type for the measurement.
Measurements are specified by x

Group  
Measurement

Syntax  

MEASurement:MEAS<x>:FROMedge {RISe|FALL|BOTH}

MEASurement:MEAS<x>:FROMedge?

Arguments  

FALL specifies the falling edge of the waveform.
RISE specifies the rising edge of the waveform.
BOTH specifies both the rising and falling edges of the waveform.

Examples  

MEASurement:MEAS1:FROMedge RISE specifies the rising edge of the
waveform.

MEASurement:MEAS1:FROMedge? might return
:MEASUREMENT:MEAS1:FROMEDGE BOTH indicating both the rising and falling
edges of the waveform.

**MEASUrement:MEAS<x>:FROMEDGESEARCHDIRect**

This command sets or queries the from edge search direction for the measurement.
Measurements are specified by x

Group  
Measurement

Syntax  

MEASurement:MEAS<x>:FROMEDGESEARCHDIRect {FORWard|BACKward}

MEASurement:MEAS<x>:FROMEDGESEARCHDIRect?
Arguments

FORWARD specifies a forward search from the edge.

BACKWARD specifies a backward search from the edge.

Examples

MEASUREMENT:MEAS1:FROMEDGESEARCHDIRECT BACKWARD specifies a backward search from the edge.

MEASUREMENT:MEAS1:FROMEDGESEARCHDIRECT? might return
:MEASUREMENT:MEAS1:FROMEDGESEARCHDIRECT FORWARD indicating a forward search from the edge.

MEASUREMENT:MEAS<x>:GATING

This command sets or queries the gating type for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

MEASUREMENT:MEAS<x>:GATING {NONE|SCREEN|CURSOR|LOGIC|SEARCH}

MEASUREMENT:MEAS<x>:GATING?

Arguments

NONE specifies measurements are taken across the entire record.

SCREEN turns on gating, using the left and right edges of the screen.

CURSOR limits measurements to the portion of the waveform between the vertical bar cursors, even if they are off screen.

LOGIC specifies that measurements are taken only when the logical state of other waveforms is true.

SEARCH specifies that measurements are taken only where the results of a user specified search are found.

Examples

MEASUREMENT:MEAS1:GATING CURSOR limits measurements to the portion of the waveform between the vertical bar cursor.


MEASUREMENT:MEAS<x>:GATING:ACTIVE

This command sets or queries the gating active level when the gating type is logic. Measurements are specified by x.
### MEASUrement:MEAS<x>:GATing:ACTive

This command sets or queries the gating settings active flag. Measurements are specified by x.

<table>
<thead>
<tr>
<th>Group</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>MEASUREMENT:MEAS&lt;x&gt;:GATING:ACTIVE {HIGH</td>
</tr>
<tr>
<td></td>
<td>MEASUREMENT:MEAS&lt;x&gt;:GATING:ACTIVE?</td>
</tr>
<tr>
<td>Arguments</td>
<td>HIGH takes a measurement when logic gating is High.</td>
</tr>
<tr>
<td></td>
<td>LOW takes a measurement when logic gating Low.</td>
</tr>
<tr>
<td>Examples</td>
<td>MEASUREMENT:MEAS1:GATING:ACTIVE LOW specifies taking measurements when logic gating is Low.</td>
</tr>
</tbody>
</table>

### MEASUrement:MEAS<x>:GATing:GLOBal

This command sets or queries the gating settings global flag. Measurements are specified by x.

<table>
<thead>
<tr>
<th>Group</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>MEASUREMENT:MEAS&lt;x&gt;:GATING:GLOBAL {OFF</td>
</tr>
<tr>
<td></td>
<td>MEASUREMENT:MEAS&lt;x&gt;:GATING:GLOBAL?</td>
</tr>
<tr>
<td>Arguments</td>
<td>OFF specifies gate settings can be changed independently for each individual measurement.</td>
</tr>
<tr>
<td></td>
<td>ON applies global gate settings to all the measurements' gate settings.</td>
</tr>
<tr>
<td></td>
<td>0 specifies gate settings can be changed independently for each individual measurement.</td>
</tr>
<tr>
<td></td>
<td>1 applies global gate settings to all the measurements' gate settings.</td>
</tr>
<tr>
<td>Examples</td>
<td>MEASUREMENT:MEAS1:GATING:GLOBAL OFF specifies gate settings can be changed independently for each individual measurement.</td>
</tr>
</tbody>
</table>
MEASUrement:MEAS<x>:GATing:HYSTeresis

This command sets or queries the gating hysteresis value when the gating type is logic. Measurements are specified by x.

Group: Measurement

Syntax:
- `MEASUrement:MEAS<x>:GATing:HYSTeresis <NR3>`
- `MEASUrement:MEAS<x>:GATing:HYSTeresis?`

Arguments:
- `<NR3>` is the gating hysteresis.

Examples:
- `MEASUrement:MEAS1:GATing:HYSTeresis 25.0E-3` sets the hysteresis to 25 mV.
- `MEASUrement:MEAS1:GATing:HYSTeresis?` might return `:MEASUREMENT:MEAS1:GATING:HYSTERESIS 30.0000E-3` indicating the hysteresis is set to 30.0 mV.

MEASUrement:MEAS<x>:GATing:LOGICSource

This command sets or queries the gating data source when the gating type is logic. The measurement number is specified by x.

Group: Measurement

Syntax:
- `MEASUrement:MEAS<x>:GATing:LOGICSource {CH<x>|MATH<x>|REF<x>}

Arguments:
- Arguments are the sources for logic gating.

Examples:
- `MEASUrement:MEAS1:GATing:LOGICSource CH3` sets the gating logic source to channel 3.
**MEASUrement:MEAS<x>:GATing:MIDRef**

This command sets or queries the gating mid ref value when the gating type is logic. Measurements are specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MEAS<x>:GATing:MIDRef <NR3>  
MEASUrement:MEAS<x>:GATing:MIDRef?

**Arguments**  
<NR3> is the mid ref value for gating.

**Examples**  
MEASUrement:MEAS1:GATing:MIDRef 1.0E+0 sets the gating midref to 1.0.  
MEASUrement:MEAS1:GATing:MIDRef? might return  
:MEASUREMENT:MEAS1:GATING:MIDREF 0.0E+0 indicating the midref value is set to 0.0.

**MEASUrement:MEAS<x>:GATing:SEARCHSource**

This command sets or queries the gating search source when the gating type is search. The measurement number is specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MEAS<x>:GATing:SEARCHSource SEARCH1

**Arguments**  
SEARCH1 is the gating source for search gating.

**Examples**  
MEASUrement:MEAS1:GATing:SEARCHSource SEARCH1 sets the gating search source to SEARCH1.  
MEASUrement:MEAS1:GATing:SEARCHSource? might return  
:MEASUREMENT:MEAS1:GATING:SEARCHSOURCE SEARCH1 indicating the gating search source is SEARCH1.

**MEASUrement:MEAS<x>:GLOBalref**

This command sets or queries the reference levels global flag for the measurement. Measurements are specified by x.
Commands listed in alphabetical order

**Group** Measurement

**Syntax**

- `MEASUREMENT:MEAS<x>:GLOBALref {OFF|ON|0|1}
- `MEASUREMENT:MEAS<x>:GLOBALref?`

**Arguments**

- OFF allows ref levels to be set separately for each measurement.
- ON applies the same ref levels to all measurements.
- 0 allows ref levels to be set separately for each measurement.
- 1 applies the same ref levels to all measurements.

**Examples**

- `MEASUREMENT:MEAS1:GLOBALref 0` allows ref levels to be set separately for each measurement.
- `MEASUREMENT:MEAS1:GLOBALref?` might return `MEASUREMENT:MEAS1:GLOBALREF 1` indicating the same ref levels apply to all measurements.

**MEASUrement:MEAS<x>:HIGHREFVoltage**

This command sets or queries the high reference voltage value for the 'time outside level' measurement. Measurements are specified by x.

**Group** Measurement

**Syntax**

- `MEASUREMENT:MEAS<x>:HIGHREFVoltage <NR3>
- `MEASUREMENT:MEAS<x>:HIGHREFVoltage?`

**Arguments**

- `<NR3>` is the high reference voltage value for the selected configuration.

**Examples**

- `MEASUREMENT:MEAS1:HIGHREFVoltage 1.5` sets the high reference voltage to 1.5 V.
- `MEASUREMENT:MEAS1:HIGHREFVoltage?` might return `MEASUREMENT:MEAS1:HIGHREFVOLTAGE 1.0000` indicating the high reference voltage is set to 1.0 V.
Commands listed in alphabetical order

**MEASUrement:MEAS<x>:IDLETime**

This command sets or queries the idle time for the measurement when the measurement type is burst width. Measurements are specified by x.

Group  
Measurement

Syntax  
MEASUrement:MEAS<x>:IDLETime <NR3>
MEASUrement:MEAS<x>:IDLETime?

Arguments  
<NR3> is the idle time.

Examples  
MEASUrement:MEAS1:IDLETime 40.0E-6 sets the idle time to 40.0 μs.
MEASUrement:MEAS1:IDLETime? might return :MEASUREMENT:MEAS1:IDLETIME 50.0000E-6 indicating the idle time is 50.0 μs.

**MEASUrement:MEAS<x>:JITTERSummary:DCD**

This command sets or queries whether DCD is included in the jitter summary for the measurement. Measurements are specified by x.

Group  
Measurement

Syntax  
MEASUrement:MEAS<x>:JITTERSummary:DCD {0|1}
MEASUrement:MEAS<x>:JITTERSummary:DCD?

Arguments  
1 add the DCD measurement as part of jitter summary.
0 do not add the DCD measurement as part of jitter summary.

Examples  
MEASUrement:MEAS1:JITTERSummary:DCD 0 specifies that the DCD measurement is not part of the jitter summary.
MEASUrement:MEAS1:JITTERSummary:DCD? might return :MEASUREMENT:MEAS4:JITTERSUMMARY:DCD 1 indicating the DCD measurement is part of the jitter summary.
**MEASUrement:MEAS<x>:JITTERSummary:DDJ**

This command sets or queries whether DDJ is included in the jitter summary for the measurement. Measurements are specified by x.

**Group** Measurement

**Syntax**

```
MEASUrement:MEAS<x>:JITTERSummary:DDJ {0|1}
MEASUrement:MEAS<x>:JITTERSummary:DDJ?
```

**Arguments**

1 add the DDJ measurement as part of jitter summary.
0 do not add the DDJ measurement as part of jitter summary.

**Examples**

MEASUrement:MEAS4:JITTERSummary:DDJ 0 specifies that the DDJ measurement is not part of the jitter summary.

MEASUrement:MEAS4:JITTERSummary:DDJ? might return :MEASUREMENT:MEAS4:JITTERSUMMARY:DDJ 1 indicating the DDJ measurement is part of the jitter summary.

**MEASUrement:MEAS<x>:JITTERSummary:DJDD**

This command sets or queries whether DJ-dd is included in the jitter summary for the measurement. Measurements are specified by x.

**Group** Measurement

**Syntax**

```
MEASUrement:MEAS<x>:JITTERSummary:DJDD {0|1}
MEASUrement:MEAS<x>:JITTERSummary:DJDD?
```

**Arguments**

1 add the DJDD measurement as part of jitter summary.
0 do not add the DJDD measurement as part of jitter summary.

**Examples**

MEASUrement:MEAS4:JITTERSummary:DJDD 0 specifies the DJDD measurement is not part of the jitter summary.

MEASUrement:MEAS4:JITTERSummary:DJDD? might return :MEASUREMENT:MEAS4:JITTERSUMMARY:DJDD 1 indicating the DJDD measurement is part of the jitter summary.
**MEASUrement:MEAS<x>:JITTERSummary:EYEWIDTHBER**

This command sets or queries whether EyeWidth@BER is included in the jitter summary for the measurement. Measurements are specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MEAS<x>:JITTERSummary:EYEWIDTHBER {0|1}  
MEASUrement:MEAS<x>:JITTERSummary:EYEWIDTHBER?

**Arguments**  
1 add the EyeWidth@BER measurement as part of jitter summary.  
0 do not add the EyeWidth@BER measurement as part of jitter summary.

**Examples**  
MEASUrement:MEAS4:JITTERSummary:EYEWIDTHBER 0 specifies that the EyeWidth@BER measurement is not part of the jitter summary.

MEASUrement:MEAS4:JITTERSummary:EYEWIDTHBER? might return :MEASUREMENT:MEAS4:JITTERSUMMARY:EYEWIDTHBER 1 indicating EyeWidth@BER is part of the jitter summary.

**MEASUrement:MEAS<x>:JITTERSummary:NPJ**

This command sets or queries whether NPJ is included in the jitter summary for the measurement. Measurements are specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MEAS<x>:JITTERSummary:NPJ {0|1}  
MEASUrement:MEAS<x>:JITTERSummary:NPJ?

**Arguments**  
1 add the NPJ measurement as part of jitter summary.  
0 do not add the NPJ measurement as part of jitter summary.

**Examples**  
MEASUrement:MEAS4:JITTERSummary:NPJ 0 specifies that the NPJ measurement is not part of the jitter summary.

MEASUrement:MEAS4:JITTERSummary:NPJ? might return :MEASUREMENT:MEAS4:JITTERSUMMARY:NPJ 1 indicating the NPJ measurement is part of the jitter summary.
MEASUrement:MEAS<x>:JITTERSummary:PJ

This command sets or queries whether PJ is included in the jitter summary for the measurement. Measurements are specified by x.

Group  Measurement
Syntax  MEASUrement:MEAS<x>:JITTERSummary:PJ {0|1}
        MEASUrement:MEAS<x>:JITTERSummary:PJ?
Arguments  1 add the PJ measurement as part of jitter summary.
        0 do not add the PJ measurement as part of jitter summary.
Examples  MEASUrement:MEAS4:JITTERSummary:PJ 0 specifies the PJ measurement is not part of jitter summary.
        MEASUrement:MEAS4:JITTERSummary:PJ? might return
        :MEASUREMENT:MEAS4:JITTERSUMMARY:PJ 1 indicating the PJ measurement is part of jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:RJDD

This command sets or queries whether RJ-dd is included in the jitter summary for the measurement. Measurements are specified by x.

Group  Measurement
Syntax  MEASUrement:MEAS<x>:JITTERSummary:RJDD {0|1}
        MEASUrement:MEAS<x>:JITTERSummary:RJDD?
Arguments  1 add the RJ-dd measurement as part of jitter summary.
        0 do not add the RJ-dd measurement as part of jitter summary.
Examples  MEASUrement:MEAS4:JITTERSummary:RJDD 0 specifies the RJ-dd measurement is not part of the jitter summary.
        MEASUrement:MEAS4:JITTERSummary:RJDD? might return
        :MEASUREMENT:MEAS4:JITTERSUMMARY:RJDD 1 indicating the RJ-dd measurement is part of the jitter summary.
Commands listed in alphabetical order

**MEASUrement:MEAS<x>:JITTERSummary:TIE**

This command sets or queries whether TIE is included in the jitter summary for the measurement. Measurements are specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MEAS<x>:JITTERSummary:TIE {0|1}  
MEASUrement:MEAS<x>:JITTERSummary:TIE?

**Arguments**  
1 add the TIE measurement as part of jitter summary.  
0 do not add the TIE measurement as part of jitter summary.

**Examples**  
MEASUrement:MEAS4:JITTERSummary:TIE 0 specifies the TIE measurement is not part of jitter summary.  

**MEASUrement:MEAS<x>:JITTERSummary:TJBER**

This command sets or queries whether TJ@BER is included in the jitter summary for the measurement. Measurements are specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MEAS<x>:JITTERSummary:TJBER {0|1}  
MEASUrement:MEAS<x>:JITTERSummary:TJBER?

**Arguments**  
1 add the TJ@BER measurement as part of jitter summary.  
0 do not add the TJ@BER measurement as part of jitter summary.

**Examples**  
MEASUrement:MEAS4:JITTERSummary:TJBER 0 specifies that the TJ@BER measurement is not part of jitter summary.  
MEASUrement:MEAS4:JITTERSummary:TJBER? might return :MEASUREMENT:MEAS4:JITTERSUMMARY:TJBER 1 indicating the TJ@BER measurement is part of jitter summary.
**MEASUrement:MEAS<x>:LABel**

This command sets or queries the label for the measurement. As the label can contain non-7-bit ASCII text, it is stored in Percent Encoding format. The measurement number is specified by x.

**Group**  Measurement

**Syntax**  MEASUrement:MEAS<x>:LABel <QString>

**Arguments**  <QString> is the measurement label.

**Examples**  
MEASUrement:MEAS1:LABel "Delay" sets the label to Delay.

MEASUrement:MEAS1:LABel? might return :MEASUREMENT:MEAS1:LABEL "Peak-to-Peak" indicating that the measurement 1 label is Peak-to-peak.

**MEASUrement:MEAS<x>:LOWREFVoltage**

This command sets or queries the low reference voltage value for the 'time outside level' measurement. Measurements are specified by x.

**Group**  Measurement

**Syntax**  MEASUrement:MEAS<x>:LOWREFVoltage <NR3>

MEASUrement:MEAS<x>:LOWREFVoltage?

**Arguments**  <NR3> is the low reference voltage value for the selected configuration.

**Examples**  
MEASUrement:MEAS1:LOWREFVoltage -1.30 sets the low reference voltage to -1.3 V.

MEASUrement:MEAS1:LOWREFVoltage? might return :MEASUREMENT:MEAS1:LOWREFVOLTAGE -1.0000 indicating the low ref voltage is -1.0 V.

**MEASurement:MEAS<x>:MAXCycle**

This command sets or queries the maximum cycle value for the DDRTERRN and DDRTERRMN measurements.
Conditions
Requires option 6-DBDDR3
Requires 6 Series MSO oscilloscope

Group
Measurements

Syntax
MEASurement:MEAS<x>:MAXCycle <NR1>
MEASurement:MEAS<x>:MAXCycle?

Arguments
<NR1> is the maximum cycle range limit value in the range or 2 to 50.

Examples
MEASurement:MEAS2:MAXCycle 45 sets the maximum cycle value for measurement 2 to 45.
MEASurement:MEAS1:MAXCycle? might return MEASurement:MEAS1:MAXCycle 17, indicating the maximum cycle value is set to 17 for measurement 1.

MEASUrement:MEAS<x>:MEASRange:GLOBal
This command sets or queries the range settings global flag for the measurement. Measurements are specified by x

Group
Measurement

Syntax
MEASurement:MEAS<x>:MEASRange:GLOBal {OFF|ON|0|1}
MEASurement:MEAS<x>:MEASRange:GLOBal?

Arguments
OFF specifies that range settings can be set independently for each individual measurement.
ON applies global measurement range settings to all the measurements' range settings.
0 specifies that range settings can be set independently for each individual measurement.
1 applies global measurement range settings to all the measurements' range settings.

Examples
MEASurement:MEAS1:MEASRange:GLOBal 1 applies global measurement range settings to all the measurements' range settings.
Commands listed in alphabetical order

MEASUrement:MEAS1:MEASRange:GLOBal? might return :MEASUREMENT:MEAS1:MEASRANGE:GLOBAL 0 indicating that range settings can be set independently for each individual measurement.

MEASUrement:MEAS<x>:MEASRange:MAX

This command sets or queries the range maximum value for the measurement. Measurements are specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:MEASRange:MAX <NR3>

MEASUrement:MEAS<x>:MEASRange:MAX?

Arguments <NR3> is the maximum measurement range limit value.

Examples MEASUrement:MEAS1:MEASRange:MAX 2.50 sets the maximum range to 2.5 V.

MEASUrement:MEAS1:MEASRange:MAX? might return :MEASUREMENT:MEAS1:MEASRANGE:MAX 1.0000 indicating the maximum range is 1.0 V.

MEASUrement:MEAS<x>:MEASRange:MIN

This command sets or queries the range minimum value for the measurement. Measurements are specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:MEASRange:MIN <NR3>

MEASUrement:MEAS<x>:MEASRange:MIN?

Arguments <NR3> is the minimum measurement range limit value.

Examples MEASUrement:MEAS1:MEASRange:MIN -1.0 sets the minimum measurement range to -1.0 V.

MEASUrement:MEAS1:MEASRange:MIN? might return :MEASUREMENT:MEAS1:MEASRANGE:MIN 0.0E+0 indicating the minimum range is 0.0 V.
Commands listed in alphabetical order

**MEASUrement:MEAS<x>:MEASRange:STATE**

This command sets or queries the range state for the measurement. Measurements are specified by x.

**Group**
Measurement

**Syntax**
MEASUrement:MEAS<x>:MEASRange:STATE {OFF|ON|0|1}
MEASUrement:MEAS<x>:MEASRange:STATE?

**Arguments**
OFF turns off the measurement range limits.
ON turns on the measurement range limits.
0 turns off the measurement range limits.
1 turns on the measurement range limits.

**Examples**
MEASUrement:MEAS1:MEASRange:STATE 0 turns off the measurement range limits.
MEASUrement:MEAS1:MEASRange:STATE? might return :MEASUREMENT:MEAS1:MEASRANGE:STATE 1 indicating the measurement range limits are on.

**MEASurement:MEAS<x>:MINCycle**

This command sets or queries the minimum cycle value for the DDRTERRN and DDRTERRMN measurements.

**Conditions**
Requires option 6-DBDDR3
Requires 6 Series MSO oscilloscope

**Group**
Measurements

**Syntax**
MEASurement:MEAS<x>:MINCycle <NR1>
MEASurement:MEAS<x>:MINCycle?

**Arguments**
<NR1> is the minimum cycle range limit value in the range or 2 to 50.
Examples

Example 1: MEASurement:MEAS2:MINCycle 20 sets the minimum cycle value for measurement 2 to 20.

Example 2: MEASurement:MEAS1:MINCycle? might return MEASurement:MEAS1:MINCycle 17, indicating the minimum cycle value is set to 17 for measurement 1.

**MEASUrement:MEAS<x>:PATTERNDETECTION**

This command sets or queries the pattern detection type for the measurement. Measurements are specified by x.

**Group**
Measurement

**Syntax**
MEASUrement:MEAS<x>:PATTERNDETECTION {AUTO|MANUAL}
MEASUrement:MEAS<x>:PATTERNDETECTION?

**Arguments**
AUTO automatically detects the pattern.
MANUAL requires manually detecting the pattern.

**Examples**
Example 1: MEASUrement:MEAS1:PATTERNDETECTION MANUAL specifies manually detecting the pattern.


**MEASUrement:MEAS<x>:PATTERNLENgth**

This command sets or queries the pattern length for the measurement. Measurements are specified by x.

**Group**
Measurement

**Syntax**
MEASUrement:MEAS<x>:PATTERNLENgth <NR3>
MEASUrement:MEAS<x>:PATTERNLENgth?

**Arguments**
<NR3> is the pattern length.
Commands listed in alphabetical order

**Examples**

```plaintext
MEAS1:_PATTERNLENGTH 3.0 sets the pattern length to 3.0.
MEAS1:_PATTERNLENGTH? might return
:MEASUREMENT:MEAS1:_PATTERNLENGTH 2.0000 indicating the
pattern length is 2.0.
```

**MEASUREMENT:MEAS<x>:PATTERNTYPE**

This command sets or queries the pattern type for the measurement. Measurements are specified by x.

**Group**  
Measurement

**Syntax**

```plaintext
MEASUREMENT:MEAS<x>:PATTERNTYPE {REPeating|ARBitrary}
MEASUREMENT:MEAS<x>:PATTERNTYPE?
```

**Arguments**

- **REPeating** specifies a repeating pattern.
- **ARBitrary** specifies an arbitrary pattern.

**Examples**

```plaintext
MEAS1:_PATTERNTYPE ARBITRARY specifies an arbitrary pattern.
MEAS1:_PATTERNTYPE? might return
:MEASUREMENT:MEAS1:_PATTERNTYPE REPEATING indicating a repeating pattern.
```

**MEASUREMENT:MEAS<x>:PERFREQ:EDGE**

This command sets or queries the edge type of a Period/Frequency measurement. The measurement number is specified by x.

**Group**  
Measurement

**Syntax**

```plaintext
MEASUREMENT:MEAS<x>:PERFREQ:EDGE {FIRST|RISE|FALL}
```

**Arguments**

- **FIRST** computes the measurement between Rising edges if the first edge is Rising.
  Computes the measurement between Falling edges if the first edge is Falling.
- **RISE** computes the measurement between Rising edges.
- **FALL** computes the measurement between Falling edges.
Examples

MEASurement:MEAS1:PERFREQ:EDGE RISE specifies computing the measurement between Rising edges.

measurement:meas1:perfreq:edge? might return :MEASUREMENT:MEAS1:PERFREQ:EDGE FIRST if the measurement is computed between edges of the first type edge found.

MEASUrement:MEAS<x>:POLarity

This command sets or queries the polarity for the measurement when the measurement type is burst width. Measurements are specified by x.

Group Measurement

Syntax

MEASUrement:MEAS<x>:POLarity {NORMal|INVerted}

MEASUrement:MEAS<x>:POLarity?

Arguments

NORMal specifies normal polarity.

INVerted specifies inverted polarity.

Examples

MEASUrement:MEAS1:POLarity INVERTED specifies inverted polarity.


MEASUrement:MEAS<x>:POPUlation:GLOBal

This command sets or queries the population settings global flag. The measurement number is specified by x.

Group Measurement

Syntax

MEASUrement:MEAS<x>:POPUlation:GLOBal {OFF|ON|0|1}

MEASUrement:MEAS<x>:POPUlation:GLOBal?

Arguments

OFF specifies that population settings can be changed independently for each individual measurement.

ON applies the global population settings to all the measurements' population settings.
Commands listed in alphabetical order

0 specifies that population settings can be changed independently for each individual measurement.

1 applies the global population settings to all the measurements' population settings.

**Examples**

```
MEASUREMENT:MEAS1:POPULATION:GLOBAL 1 applies the global population settings to all the measurements' population settings.

MEASUREMENT:MEAS1:POPULATION:GLOBAL? might return :MEASUREMENT:MEAS1:POPULATION:GLOBAL 0 indicating that population settings can be changed independently for each individual measurement.
```

**MEASUrement:MEAS<x>:POPUlation:LIMIT:STATE**

This command sets or queries the population limit state for the measurement. The measurement number is specified by x.

**Group** Measurement

**Syntax**

```
MEASUREMENT:MEAS<x>:POPULATION:LIMIT:STATE {OFF|ON|0|1}
MEASUREMENT:MEAS<x>:POPULATION:LIMIT:STATE?
```

**Arguments**

- OFF turns off the population limit.
- ON turns on the population limit.
- 0 turns off the population limit.
- 1 turns on the population limit.

**Examples**

```
MEASUREMENT:MEAS1:POPULATION:LIMIT:STATE 0 turns off the population limit.

```

**MEASUrement:MEAS<x>:POPUlation:LIMIT:VALue**

This command sets or queries the population limit value for the measurement. The measurement number is specified by x.

**Group** Measurement
Commands listed in alphabetical order

**MEASurement:MEAS<x>:POPUlation:LIMIT:VALue**

**Syntax**

```
MEASurement:MEAS<x>:POPUlation:LIMIT:VALue <NR3>
MEASurement:MEAS<x>:POPUlation:LIMIT:VALue?
```

**Arguments**

<NR3> the current limit value.

**Examples**

```
```

**MEASUrement:MEAS<x>:REFLevels:ABSolute:FALLHigh**

This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute. Measurements are specified by x.

**NOTE. This command affects the results of rise and fall measurements**

**Group**

Measurement

**Syntax**

```
MEASUREMENT:MEAS<x>:REFLevels:ABSolute:FALLHigh <NR3>
MEASUREMENT:MEAS<x>:REFLevels:ABSolute:FALLHigh?
```

**Arguments**

<NR3> is the high reference level in volts. The default is 0.0 V.

**Examples**

```
MEASUREMENT:MEAS1:REFLEVELS:ABSOLUTE:FALLHIGH 1.71 sets the high reference level to 1.71 V.
MEASUREMENT:MEAS1:REFLEVELS:ABSOLUTE:FALLHIGH? might return :MEASUREMENT:MEAS1:REFLEVELS:ABSOLUTE:FALLHIGH 1.7100E+00, indicating that the absolute high reference level is set to 1.71 V.
```

**MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLLow**

This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute. Measurements are specified by x.

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer 2-525
NOTE. This command affects the results of rise and fall measurements.

Group  
Measurement

Syntax  
MEASurement:MEAS<x>:REFLevels<x>:ABSolute:FALLLow <NR3>  
MEASurement:MEAS<x>:REFLevels<x>:ABSolute:FALLLow?

Arguments  
<NR3> is the low reference level in volts. The default is 0.0 V.

Examples  
MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLLOW 0.0 V sets the low reference level to 0.0 V.

MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLLOW? might return :MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLLOW 0.0000E+00, indicating that the absolute low reference level is set to 0.0 V.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLMid

This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute. Measurements are specified by x.

NOTE. This command affects the results of period, frequency, delay, and all cyclic measurements.

Group  
Measurement

Syntax  
MEASurement:MEAS<x>:REFLevels<x>:ABSolute:FALLMid <NR3>  
MEASurement:MEAS<x>:REFLevels<x>:ABSolute:FALLMid?

Arguments  
<NR3> is the mid reference level in volts. The default is 0.0 V.

Examples  
MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLMID 0.5 sets the mid reference level for the delay waveform to 0.5 V.

MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLMID? might return :MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLMID 0.5000E+00, indicating that the absolute mid reference level is set to 0.5 V.
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:HYSteresis

This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute. The measurement number is specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:HYSteresis <NR3>

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:HYSteresis?

Arguments <NR3> is the hysteresis value used for autoseet.

Examples MEASUrement:MEAS1:REFLevels1:ABSolute:HYSteresis 25.0E-3 sets the hysteresis to 25 mV.

MEASUrement:MEAS1:REFLevels1:ABSolute:HYSteresis? might return :MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:HYSTERESIS 30.0000E-3 indicating the hysteresis is set to 30 mV.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEHigh

This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute. The measurement number is specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEHigh <NR3>

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEHigh?

Arguments <NR3> is the high reference level, and is the zero percent level when the measurement's Ref level method is set to Absolute.

Examples MEASUrement:MEAS1:REFLevels1:ABSolute:RISEHigh 1.50 sets the high reference level to 1.5 V.

MEASUrement:MEAS1:REFLevels1:ABSolute:RISEHigh? might return :MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISEHIGH 1.000000 indicating the high reference level is set to 1.0 V.
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISELow

This command sets or queries the value used as the low reference level of the rising edge when the measurement's Ref level method is set to absolute. The measurement number is specified by x.

Syntax

MEASUrement:MEAS<x>:REFlevels<x>:ABSolute:RISELow <NR3>
MEASUrement:MEAS<x>:REFlevels<x>:ABSolute:RISELow?

Arguments

<NR3> is the low reference level, and is the zero percent level when the measurement's Ref level method is set to Absolute.

Examples

MEASUrement:MEAS1:REFLevels1:ABSolute:RISELow -1.50 sets the low reference level to -1.5 V.
MEASUrement:MEAS1:REFLevels1:ABSolute:RISELow? might return :MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISELOW -1.0000 indicating the low reference level is -1.0 V.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEMId

This command sets or queries the value used as the mid reference level of the rising edge when the measurement's Ref level method is set to absolute. The measurement number is specified by x.

Syntax

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEMid <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEMid?

Arguments

<NR3> is the mid reference level (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Absolute.

Examples

MEASUrement:MEAS1:REFLevels1:ABSolute:RISEMid 30.0E-3 sets the mid reference level to 30 mV.
MEASUrement:MEAS1:REFLevels1:ABSolute:RISEMid? might return :MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISEMID 0.0E+0 indicating the mid reference level is 0.0 V.
**MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:TYPE**

This command sets or queries the reference level type for the measurement. The measurement number is specified by x.

**Group** Measurement

**Syntax**

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:TYPE {SAME|UNIQue}

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:TYPE?

**Arguments**

SAME specifies that the absolute levels are set the same.

UNIQue specifies that the absolute levels can be set independently.

**Examples**

MEASUrement:MEAS1:REFLevels1:ABSolute:TYPE UNIQUE specifies that the absolute levels can be set independently.


**MEASUrement:MEAS<x>:REFLevels<x>:BASETop**

This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the measurement. The measurement number is specified by x.

**Group** Measurement

**Syntax**

MEASUrement:MEAS<x>:REFLevels<x>:BASETop

{AUTO|MINMax|MEANhistogram|MODEhistogram|EYEhistogram}

MEASUrement:MEAS<x>:REFLevels<x>:BASETop?

**Arguments**

AUTO automatically chooses a reference level method.

MINMax specifies that reference levels are relative to the measurement MIN and MAX.

MEANhistogram specifies that reference levels are relative to the histogram mean BASE and TOP.

MODEhistogram specifies that reference levels are relative to the histogram mode BASE and TOP.
EYE histogram specifies that reverence levels are relative to the eye histogram BASE and TOP.

**Examples**

- `MEASurement:MEAS1:REFLevels1:BASETop MINMAX` specifies that reference levels are relative to the measurement MIN and MAX.

### MEASUrement:MEAS<x>:REFLev els<x>:METHOD

This command sets or queries the method used to calculate reference levels for the measurement. The measurement number is specified by `x`.

**Group** Measurement

**Syntax**

```plaintext
MEASurement:MEAS<x>:REFLev els<x>:METHOD {PERCent|ABSolute}
MEASurement:MEAS<x>:REFLev els<x>:METHOD?
```

**Arguments**

- **PERCent** specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the `MEASurement:MEAS<x>:REFLevel:PERCent` commands.
- **ABSolute** specifies that the reference levels are set explicitly using the `MEASurement:MEAS<x>:REFLevel:ABSolute` commands. This method is useful when precise values are required.

**Examples**

- `MEASUREMENT:MEAS1:REFLEVELS1:METHOD ABSOLUTE` specifies that explicit user-defined values are used for the reference levels.
- `MEASUREMENT:MEAS1:REFLEVELS1:METHOD?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:METHOD PERCENT`, indicating that the reference level units used are calculated as a percent relative to HIGH and LOW.

### MEASUrement:MEAS<x>:REFLev els<x>:PERCent:FALLHigh

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent. The measurement number is specified by `x`.

2-530 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
Commands listed in alphabetical order

**Group**: Measurement

**Syntax**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MEASUREMENT:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:FALLHigh &lt;NR3&gt;</code></td>
<td>Sets or queries the percentage (where 100% is equal to HIGH) used to calculate the high reference level.</td>
</tr>
<tr>
<td><code>MEASUREMENT:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:FALLHigh?</code></td>
<td>Retrieves the high reference level.</td>
</tr>
</tbody>
</table>

**Arguments**

- `<NR3>` is the percentage (where 100% is equal to HIGH) used to calculate the high reference level.

**Examples**

- `MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLHIGH 95.0` sets the high reference level of the falling edge to 95%.
- `MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLHIGH?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLHIGH 90.0000` indicating the high reference level is set to 90%.

**MEASUREMENT:MEAS<x>:REFLevels<x>:PERCent:FALLLow**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement's ref level method is set to percent. The measurement number is specified by `x`.

**Group**: Measurement

**Syntax**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MEASUREMENT:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:FALLLow &lt;NR3&gt;</code></td>
<td>Sets or queries the percentage (where 100% is equal to HIGH) used to calculate the low reference level.</td>
</tr>
<tr>
<td><code>MEASUREMENT:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:FALLLow?</code></td>
<td>Retrieves the low reference level.</td>
</tr>
</tbody>
</table>

**Arguments**

- `<NR3>` is the percentage (where 100% is equal to HIGH) used to calculate the low reference level.

**Examples**

- `MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLLOW 5.0` sets the low reference level of the falling edge to 5%.

**MEASUREMENT:MEAS<x>:REFLevels<x>:PERCent:FALLMid**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge.
when the measurement's ref level method is set to percent. The measurement number is specified by x.

**Group** Measurement

**Syntax**

<table>
<thead>
<tr>
<th>Command 1</th>
<th>Command 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MEASUREMENT:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:FALLMid &lt;NR3&gt;</code></td>
<td><code>MEASUREMENT:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:FALLMid?</code></td>
</tr>
</tbody>
</table>

**Arguments**

<NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level.

**Examples**

- `MEASUREMENT:MEAS1:REFLevels1:PERCent:FALLMid 50.0` sets the mid reference level of the falling edge to 50%.
- `MEASUREMENT:MEAS1:REFLevels1:PERCent:FALLMid?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLMID 50.0000` indicating the mid reference level of the falling edge is set to 50%.

**MEASUREMENT:MEAS<x>:REFLevels<x>:PERCent:HYSTEResis**

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent. The measurement number is specified by x.

**Group** Measurement

**Syntax**

<table>
<thead>
<tr>
<th>Command 1</th>
<th>Command 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MEASUREMENT:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:HYSTEResis &lt;NR3&gt;</code></td>
<td><code>MEASUREMENT:MEAS&lt;x&gt;:REFLevels&lt;x&gt;:PERCent:HYSTEResis?</code></td>
</tr>
</tbody>
</table>

**Arguments**

<NR3> is the hysteresis value used for the autoset.

**Examples**

- `MEASUREMENT:MEAS1:REFLevels1:PERCent:HYSTEResis 5.0` sets the hysteresis level to 5%.
- `MEASUREMENT:MEAS1:REFLevels1:PERCent:HYSTEResis?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:HYSTERESIS 5.0000` indicating the hysteresis is 5.0%.
**MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEHigh**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

**Group** Measurement

**Syntax**

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEHigh <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEHigh?

**Arguments**

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

**Examples**

MEASUrement:MEAS1:REFLevels1:PERCent:RISEHigh 95.0 sets the high reference level of the rising edge is set to 95%.

MEASUrement:MEAS1:REFLevels1:PERCent:RISEHigh? might return :MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISEHIGH 90.0000 indicating the high reference level of the rising edge is set to 90%.

**MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISELow**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

**Group** Measurement

**Syntax**

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISELow <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISELow?

**Arguments**

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

**Examples**

MEASUrement:MEAS1:REFLevels1:PERCent:RISELow 5.0 sets the low reference level is set to 5.0%.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISELow?

might return

:MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISELOW 10.0000
indicating the low reference level is set to 10.0%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEMid

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEMid <NR3>

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEMid?

Arguments <NR3> the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement ref level method is set to Percent.

Examples

MEASUrement:MEAS1:REFLevels1:PERCent:RISEMid 50.0000 sets the mid reference level of the rising edge is set to 50.0%.

MEASUrement:MEAS1:REFLevels1:PERCent:RISEMid? might return

:MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISEMID 50.0000 indicating the mid reference level of the rising edge is set to 50.0%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:TYPE

This command specifies or queries the reference level percent type for the measurement. The measurement number is specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:REFLevels<x>:PERCent:TYPE {TENNinety|TWENtyeighty|CUSTom}

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:TYPE?

Arguments TENNinety sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.
Commands listed in alphabetical order

TWENtyeighty sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

CUSTom allows setting other reference level percents.

Examples
MEASurement:MEAS1:REFLeveis1:PERCent:TYPE TWENtyeighty sets the reference levels percent to 20%, 50% and 80%.
MEASurement:MEAS1:REFLeveis1:PERCent:TYPE? might return :MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:TYPE TENNINETY indicating the reference levels percent type is 10%, 50% and 90% respectively.

MEASUrement:MEAS<x>:REFMode
This command sets or queries the reference level mode for the measurement. The measurement number is specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:REFMode {AUTO|MANual}
MEASUrement:MEAS<x>:REFMode?

Arguments AUTO sets the reference level for the measurement automatically.
MANual allows the user to set the reference level for the measurement.

Examples MEASUrement:MEAS1:REFMode MANual allows the user to set the reference level for the measurement.
MEASUrement:MEAS1:REFMode? might return :MEASUREMENT:MEAS1:REFMODE AUTO indicating the reference levels for the measurement are set automatically.

MEASUrement:MEAS<x>:REFVoltage
This command sets or queries the reference voltage value for the measurement. The measurement number is specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:REFVoltage <NR3>
MEASUrement:MEAS<x>:REFVoltage?

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
Arguments  

<NR3> is the reference voltage value for the selected configuration.

Examples  

MEASUrement:MEAS1:REFVoltage 30.E-3 sets the reference voltage value for the measurement to 30 mV.

MEASUrement:MEAS1:REFVoltage? might return :MEASUREMENT:MEAS1:REFVOLTAGE 0.0E+0 indicating the reference voltage value for the measurement is 0.0 V.

**MEASUrement:MEAS<x>:RESUIts:ALLAcqs:MAXimum? (Query Only)**

This query-only command returns the maximum value for all accumulated measurement acquisitions of the specified measurement. The measurement number is specified by x.

Group  
Measurement

Syntax  
MEASUrement:MEAS<x>:RESUIts:ALLAcqs:MAXimum?

Returns  
The maximum value for all accumulated measurement acquisitions of the specified measurement.

Examples  

**MEASUrement:MEAS<x>:RESUIts:ALLAcqs:MEAN? (Query Only)**

This query-only command returns the mean value for all accumulated measurement acquisitions for measurement <x>.

Group  
Measurement

Syntax  
MEASUrement:MEAS<x>:RESUIts:ALLAcqs:MEAN?

Returns  
The mean value for all accumulated measurement acquisitions for measurement <x>.
MEASUrement:MEAS<x>:RESUItS:ALLAcqs:MINimum? (Query Only)

This query-only command returns the minimum value for all accumulated measurement acquisitions for measurement <x>.

Group Measurement

Syntax MEASUrement:MEAS<x>:RESUItS:ALLAcqs:MINimum?

Returns The minimum value for all accumulated measurement acquisitions for measurement <x>.

Examples MEASUrement:MEAS1:RESUItS:ALLAcqs:MINimum? might return :MEASUREMENT:MEAS1:RESULTS:ALLACQS:MINIMUM 2.200 indicating the measurement minimum value is 2.200 V.

MEASUrement:MEAS<x>:RESUItS:ALLAcqs:PK2PK? (Query Only)

This query-only command returns the peak-to-peak value for all accumulated measurement acquisitions for measurement <x>.

Group Measurement

Syntax MEASUrement:MEAS<x>:RESUItS:ALLAcqs:PK2PK?

Returns The peak-to-peak value for all accumulated measurement acquisitions for measurement <x>.

Examples MEASUrement:MEAS1:RESUItS:ALLAcqs:PK2PK? might return :MEASUREMENT:MEAS1:RESULTS:ALLACQS:PK2PK 200.0E-3 indicating the measurement peak-to-peak value is 200 mV.
MEASUrement:MEAS<x>:RESULTs:ALLAcqs:POPUlation? (Query Only)

This query-only command returns the population measurement value for measurement <x>.

Group  Measurement
Syntax  MEASUrement:MEAS<x>:RESULTs:ALLAcqs:POPUlation?
Returns  The population measurement value for measurement <x>.

MEASUrement:MEAS<x>:RESULTs:ALLAcqs:STDDev? (Query Only)

This query-only command returns the standard deviation for all accumulated measurement acquisitions for measurement <x>.

Group  Measurement
Syntax  MEASUrement:MEAS<x>:RESULTs:ALLAcqs:STDDev?
Returns  The standard deviation for all accumulated measurement acquisitions for measurement <x>.
Examples  MEASUrement:MEAS1:RESULTs:ALLAcqs:STDDev? might return :MEASUREMENT:MEAS1:RESULTs:ALLAcqs:STDDev 23.5741246494459E-3 indicating the measurement standard deviation is 23.57 mV.

MEASUrement:MEAS<x>:RESULTs:CURRENTacq:MAXimum? (Query Only)

This query-only command returns the maximum value found for the specified measurement since the last statistical reset. The measurement number is specified by x.

Group  Measurement
Syntax  MEASUrement:MEAS<x>:RESULTS:CURRentacq:MAXimum?

Returns  The maximum value found for the specified measurement since the last statistical reset.

Examples  MEASUrement:MEAS1:RESULTS:CURRentacq:MAXimum? might return  :MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MAXIMUM 2.280 indicating the current measurement maximum value is 2.28 V.

MEASUrement:MEAS<x>:RESULTS:CURRentacq:MEAN? (Query Only)

This query-only command returns the mean value for the measurement for the current acquisition.

Group  Measurement

Syntax  MEASUrement:MEAS<x>:RESULTS:CURRentacq:MEAN?

Arguments  MEAS<x> is the measurement number for which to return a value. This is the equivalent of the number shown in the measurement badge on the UI.

Returns  The mean value accumulated for the specified measurement since the last statistical reset.

Examples  MEASUrement:MEAS3:RESULTS:CURRentacq:MEAN? might return  :MEASUREMENT:MEAS3:RESULTS:CURRENTACQ:MEAN 2.30 indicating the mean of measurement 3 in the current acquisition is 2.3 V.

MEASUrement:MEAS<x>:RESULTS:CURRentacq:MINimum? (Query Only)

This query-only command returns the minimum value found for the specified measurement since the last statistical reset. The measurement number is specified by x.

Group  Measurement

Syntax  MEASUrement:MEAS<x>:RESULTS:CURRentacq:MINimum?
Returns  The minimum value found for the specified measurement since the last statistical reset.

Examples  MEASurement:MEAS1:RESULTS:CURRentacq:MINimum? might return :MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MINIMUM 2.260 indicating the minimum value of the measurement in the current acquisition is 2.26 V.

MEASUrement:MEAS<x>:RESUIts:CURRentacq:PK2PK? (Query Only)

This query-only command returns the peak-to-peak value for the specified measurement for the current acquisition. The measurement number is specified by x.

Group  Measurement

Syntax  MEASUrement:MEAS<x>:RESUIts:CURRentacq:PK2PK?

Returns  The peak-to-peak value for the specified measurement.

Examples  MEASUrement:MEAS1:RESUIts:CURRentacq:PK2PK? might return :MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:PK2PK 0.0E+0 indicating the peak-to-peak value for the specified measurement for the current acquisition is 0.0 V.

MEASUrement:MEAS<x>:RESUIts:CURRentacq:POPUlation? (Query Only)

This query-only command returns the population for the specified measurement for the current acquisition. The measurement number is specified by x.

Group  Measurement

Syntax  MEASUrement:MEAS<x>:RESUIts:CURRentacq:POPUlation?

Returns  The population measurement value for the specified measurement.

MEASUREment:MEAS<x>:RESUlts:CURREn tácq:STDDev? (Query Only)

This query-only command returns the standard deviation for the specified measurement for all acquisitions accumulated since statistics were last reset. The measurement number is specified by x.

Group Measurement

Syntax MEASUREment:MEAS<x>:RESUlts:CURREn tácq:STDDev?

Returns The standard deviation of values accumulated for the specified measurement since the last statistical reset.

Examples MEASUREment:MEAS<x>:RESUlts:CURREn tácq:STDDev? might return :MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:STDDEV 0.0E+0 indicating the standard deviation for the measurement is 0.0 V.

MEASUREment:MEAS<x>:SIGNALType

This command sets or queries the signal type of source 1 for the measurement. The measurement number is specified by x.

Group Measurement

Syntax MEASUREment:MEAS<x>:SIGNALType {CLOCK|DATA|AUTO}
MEASUREment:MEAS<x>:SIGNALType?

Arguments CLOCK specifies a clock signal type.
DATA specifies a data signal type.
AUTO automatically selects the signal type.

Examples MEASUREment:MEAS1:SIGNALType CLOCK specifies a clock signal type.
MEASUREment:MEAS1:SIGNALType? might return :MEASUREMENT:MEAS1:SIGNALTYPE AUTO indicating the instrument automatically selects the signal type.
**MEASUrement:MEAS<x>:SOURce<x>**

This command sets or queries the measurement source. The measurement number and source are specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MEAS<x>:SOURce<x>

**Arguments**

**Examples**

**MEASUrement:MEAS<x>:SSC:Nominalfreq**

This command sets or queries the user-defined frequency for the measurement when the measurement type is SSC. The measurement number is specified by x.

**Group**  
Measurement

**Syntax**  
MEASUrement:MEAS<x>:SSC:Nominalfreq <NR3>  
MEASUrement:MEAS<x>:SSC:Nominalfreq?

**Arguments**  
<NR3> is the user-defined nominal frequency type for SSC configurations.

**Examples**  
MEASUrement:MEAS1:SSC:Nominalfreq 2.0E+9 sets the frequency to 2.0 GHz.  

**MEASUrement:MEAS<x>:SSC:Nominalfreq:SELECTIONtype**

This command sets or queries the frequency detection type for the measurement when the measurement type is SSC. The measurement number is specified by x.

**Group**  
Measurement
Commands listed in alphabetical order

**MEASurement:MEAS<x>:SSC:NOMinalfreq:SELECTIONtype**

Syntax

```
MEASurement:MEAS<x>:SSC:NOMinalfreq:SELECTIONtype {AUTO|MANual}
MEASurement:MEAS<x>:SSC:NOMinalfreq:SELECTIONtype?
```

Arguments

- **AUTO** automatically sets the detection type.
- **MANual** specifies the manual detection type.

Examples

```
MEASurement:MEAS1:SSC:NOMinalfreq:SELECTIONtype AUTO specifies the auto-detection type.
```

**MEASurement:MEAS<x>:TCKAVG**

This command sets or queries the average clock period value used in DDR measurements.

Syntax

```
MEASurement:MEAS<x>:TCKAVG {EACHCLOCKCYCLE|TWOCLOCKCYCLES}
MEASurement:MEAS<x>:TCKAVG?
```

Related Commands

- **MEASurement MEAS<x>:TIMINGMode**

Arguments

- **EACHCLOCKCYCLE** sets the DDR Timing Mode to use each clock cycle at a time.
- **TWOCLOCKCYCLES** sets the DDR Timing Mode to use two cycles at a time.

Examples

```
MEASurement:MEAS1:TIMINGMode EACHCLOCKCYCLE sets the Timing Mode to consider each clock cycle.
MEASurement:MEAS1:TIMINGMode? might return 
MEASurement:MEAS1:TIMINGMode EACHCLOCKCYCLE, indicating that Timing Mode is set to EACHCLOCKCYCLE.
```

**MEASurement:MEAS<x>:TIMINGMode**

This command sets or queries the Timing mode for the specified DDR measurement.
Commands listed in alphabetical order

**Group** Measurement

**Syntax**

```
MEASurement:MEAS<x>:TIMINGMode <NR3>
MEASurement:MEAS<x>:TIMINGMode?
```

**Related Commands**

```
MEASUrement MEAS<x>:TCKAVG
```

**Arguments**

NR3 is a floating point number that represents the DDR average clock period in seconds.

**Examples**

```
MEASurement:MEAS7:TCKAVG 2.5E-9 sets the average clock period value as 3.2 ns for measurement 7.
MEASurement:MEAS2:TCKAVG? might return MEASurement:MEAS2:TCKAVG 2.0E-9 indicating that average clock period is set to 2.0 ns for measurement 2.
```

**MEASUrement:MEAS<x>:TOEdge**

This command sets or queries the 'to edge' type for the measurement. The measurement number is specified by x.

**Group** Measurement

**Syntax**

```
MEASUrement:MEAS<x>:TOEdge
{SAMEas|OPPositeas|RISe|FALL|BOTH}
MEASUrement:MEAS<x>:TOEdge?
```

**Arguments**

FALL specifies the falling edge of the waveform.
RISE specifies the rising edge of the waveform.
BOTH specifies both a rising and falling edge of the waveform.
SAMEas specifies that both edges of the waveform are the same.
OPPositeas specifies that the edges of the waveform are not the same.

**Examples**

```
MEASUrement:MEAS1:TOEdge FALL specifies the to edge is the falling edge of the waveform.
MEASUrement:MEAS1:TOEdge? might return :MEASUREMENT:MEAS1:TOEDGE SAMEAS indicating that both edges of the waveform are the same.
```
**MEASUrement:MEAS<x>:TOEDGESEARCHDIREc**

This command sets or queries the to edge search direction for the measurement. The measurement number is specified by x.

**Group** Measurement

**Syntax**

```
MEASUrement:MEAS<x>:TOEDGESEARCHDIREc {FORWard|BACKWard}
MEASUrement:MEAS<x>:TOEDGESEARCHDIREc?
```

**Arguments**

- **FORWard** specifies a forward search to the edge.
- **BACKWard** specifies a backward search to the edge.

**Examples**

```
MEASUrement:MEAS1:TOEDGESEARCHDIREc FORWARD specifies a forward search to the edge.
MEASUrement:MEAS1:TOEDGESEARCHDIREc? might return :MEASUREMENT:MEAS1:TOEDGESEARCHDIREc FORWARD indicating the instrument will search in the forward direction for the to edge.
```

**MEASUrement:MEAS<x>:TRANSition**

This command sets or queries the transition edges flag for the measurement. The measurement number is specified by x.

**Group** Measurement

**Syntax**

```
MEASUrement:MEAS<x>:TRANSition {<NR1>|OFF|ON}
```

**Arguments**

- **<NR1>** = 1, the measurement is computed on rising (if measurement type is rise time) or falling edges (if measurement type is fall time) following a double transition only. If it is set to 0, the measurement is computed on all rising (if measurement type is rise time) or falling (if measurement type is fall time) edges.
- **OFF** computes the measurement on all rising (if measurement type is rise time) or falling (if measurement type is fall time) edges.
- **ON** computes the measurement on rising (if measurement type is rise time) or falling edges (if measurement type is fall time) following a double transition only.
Examples

MEASurement:MEAS1:TRANSition 1 specifying the measurement is computed on rising (if measurement type is rise time) or falling edges (if measurement type is fall time) following a double transition only.

MEASurement:MEAS1:TRANSition? might return :MEASUREMENT:MEAS1:TRANSITION 0 indicating the measurement is computed on all rising (if measurement type is rise time) or falling (if measurement type is fall time) edges.

MEASUrement:MEAS<x>:TYPe

This command sets or queries the measurement type for the measurement specified by x.

Group
Measurement

Syntax

MEASurement:MEAS<x>:TYPe

{ACCOMMONMODE|ACRMS|AMPLITUDE|AREA|BASE|BITAMPLITUDE|BITHIGH|BITLOW|BUSTWIDTH|COMMONMODE|DATARATE|DCD|DDJ|DDRAOS|DDRAOSPERTCK|DDRAOSPERUI|DDRAUS|DDRAUSPERTCK|DDRAUSPERUI|DDRHOLDDIFF|DDRSUPDUPDIFF|DDRTCHABS|DDRTCHAVE|DDRTCKAV|DDRTCLABS|DDRTCLAVERAGE|DDRTERRM|DDRTERRN|DDRTJITCC|DDRTJITDUTY|DDRTJITPER|DDRTTP|DDRTTPRE|DDRTWPRE|DDRVIXAC|DDTDQSCK|DELAY|DJ|DJDIRAC|EYEHIGH|EYELOW|FALLSLEWRATE|FALLTIME|FREQUENCY|F2|F4|F8|HIGH|HEIGHT|HEIGHTBER|HIGHTIME|HOLD|JITTERSUMMARY|J2|J9|LOW|LOWTIME|MAXIMUM|MEAN|MINIMUM|NDUTY|NPERIOD|NPJ|NOVERSHEED|NWIDTH|PDUTY|PERIOD|PHASE|PHASENOISE|PJ|PK2PK|POVERSHOOT|PWIDT|QFACTOR|RISESLEWRATE|RISETIME|RJ|RJDIRAC|RMS|SRJ|SSCFREQDEV|SSCMODRATE|SUP|SKEW|TIE|TIMEOUTSIDELEVEL|TJBER|TNTRATIO|TOP|UNITINTERVAL|VDIFFXOVR|WIDTH|WIDTHBER}

MEASurement:MEAS<x>:TYPe?

Arguments

ACCOMMONMODE AC Common Mode (Pk-Pk) is the peak-to-peak of the common mode voltage of two sources. This measurement is made across the entire record.

ACRMS (AC RMS) is the true Root Mean Square of the data points, about the Mean. This measurement can be made across the entire record, or on each cycle in the record.

AMPLITUDE is the difference between the Top value and the Base value. This measurement can be made across the entire record, or on each cycle in the record.

Amplitude = High - Low

AREA is the area under the curve, calculated by integrating the data points. The area measured above ground is positive. The area measured below ground is...
negative. This measurement can be made across the entire record, or on each cycle in the record.

**BASE** is the most common data value below the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

**BITAMPLITUDE** (Bit Amplitude) is the difference between the amplitudes of the 1 bit and the 0 bit surrounding a transition. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each transition bit in the record (Mean) or across the entire record (Mode).

**BITHIGH** (Bit High) is the amplitude of a 1 bit. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each high bit in the record (Mean) or across the entire record (Mode).

**BITLOW** (Bit Low) is the amplitude of a 0 bit. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each high bit in the record (Mean) or across the entire record (Mode).

**BURSTWIDTH** (Burst Width) is the duration of a series of adjacent crossings of the Mid reference level (RM). Bursts are separated by a user-defined idle time (tI). This measurement is made on each burst in the record.

**COMMONMODE** (DC Common Mode) is the arithmetic mean of the common mode voltage of two sources. This measurement is made across the entire record.

**DATARATE** (Data Rate) is the reciprocal of Unit Interval. This measurement is made on each bit in the record.

**DCD** (duty cycle distortion) is the peak-to-peak amplitude of the component of the deterministic jitter correlated with the signal polarity. This measurement is made across the entire record.

**DDJ** (data dependent jitter) is the peak-to-peak amplitude of the component of the deterministic jitter correlated with the data pattern in the waveform. This measurement is made across the entire record.

**DDRAOS** (area above signal) is the total area of the signal above a specified reference level. This measurement is made across the entire record.

**DDRAOSPERTCK** (area over signal for tCK events) is the total area of the signal above a specified reference level calculated over consecutive tCK intervals. It is applicable to clock and address/command waveforms.

**DDRAOSPERUI** (area over signal for UI events) is the total area of the signal above a specified reference level calculated over consecutive Unit intervals. It is applicable to data and data strobe waveforms.

**DDRRAUS** (area under signal) is the total area of the signal below a specified reference level. This measurement is made across the entire record.
Commands listed in alphabetical order

**DDRAUSPERTCK** (area under signal for tCK events) is the total area of the signal below a specified reference level calculated over consecutive tCK intervals. It is applicable to clock and address/command waveforms.

**DDRAUSPERUI** (area under signal for UI events) is the total area of the signal below a specified reference level calculated over consecutive unit intervals. It is applicable to data and data strobe waveforms.

**DDRHOLODDIFF** (hold difference) is the elapsed time between the specified edge of a single-ended clock waveform and the specified edge of a differential data waveform. The measurement uses the closest respective waveform edges that fall within the range limits. This measurement is made across the entire record.

**DDRSETUPDIFF** (setup difference) is the elapsed time between the specified edge of a single-ended clock waveform and when the specified edge of a differential data waveform crosses a specified level. The measurement uses the closest respective waveform edges that fall within the range limits. This measurement is made across the entire record.

**DDRTCHABS** (absolute high pulse width) is the absolute value of the high pulse width as measured from one rising edge to the next falling edge.

**DDRTCAVERAGE** (average high pulse width) is the average value of the high pulse width as measured from one rising edge to the next falling edge, across 200 consecutive cycles. This measurement is made across the entire record.

**DDRTCKAVERAGE** (average clock period) is the average clock period calculated from rising edge to rising edge, across 200 consecutive cycles. This measurement is made across the entire record.

**DDRTCLABS** (absolute low pulse width) is the absolute value of the low pulse width as measured from a falling edge to the next rising edge. This measurement is made across the entire record.

**DDTCLAVERAGE** (average low pulse width) is the average value of the low pulse width as measured from one falling edge to the next rising edge, across 200 consecutive cycles.

**DDRTERRMN** (cumulative error) is the cumulative error across multiple consecutive defined cycles from tCK(avg).

**DDRTERNN** (cumulative error) is the cumulative error across specified consecutive cycles from tCK(avg). In other words, this measures the time difference between the sum of the clock period from a 200 cycle window and n times tCK(avg).

**DDRTJITCC** (cycle to cycle jitter period) is the absolute difference in clock period between two consecutive clock cycles. This measurement is made across the entire record.

**DDRTJITDUTY** (half period jitter) is the largest elapsed time between tCH and tCH(avg), and tCL and tCL(avg), over 200 consecutive cycles.
DDRTJITPER (clock period jitter) is the largest deviation of any tCK signal from tCK(avg). This measurement is made across the entire record.

DDRTPSST (read/write burst postamble) is the width of the Read or Write burst postamble, measured from the last falling edge of the mid reference level to the start of an undriven state. This measurement is made across the entire record.

DDRTPRE (read burst preamble) is the width of the Read burst preamble, measured from exiting tristate levels to the first driving edge of the differential strobe. This measurement is made across the entire record.

DDRTWPRE is the width of the Write burst preamble, measured from exiting tristate levels to the first driving edge of the differential strobe. This measurement is made across the entire record.

DDRVIXAC is the differential input cross-point voltage measured from the true state transition (and it’s compliment) to a specified reference level, measured on a single-ended signal.

DDRTDQSCK is the strobe output access time, measured between the rising edge of the clock and before or after the differential strobe Read preamble time. Signal edges are determined by the mid-ref threshold level settings.

DELAY is the time between the specified Mid reference level (RM) crossing on one source to a specified Mid reference level (RM) crossing on a second source. This measurement is made on the first occurrence in the record.

DJ (deterministic jitter) is the peak-to-peak amplitude of all timing errors that exhibit deterministic behavior. This measurement is made across the entire record.

DJDIRAC (dual-dirac deterministic jitter) is deterministic jitter based on a simplifying assumption that the histogram of all deterministic jitter can be modeled as a pair of equal-magnitude Dirac functions. This measurement is made across the entire record.

EYEHIGH (Eye High) is the amplitude of a high (1) bit measured at a user specified location within the recovered unit interval. This measurement is made on each high bit in the record.

EYELOW (Eye Low) is the amplitude of a low (0) bit measured at a user specified location within the recovered unit interval. This measurement is made on each low bit in the record.

FALLSLEWRATE (Falling Slew Rate) is the rate of change in voltage as an edge transitions from the Top reference level (RT) to the Bottom reference level (RB). This measurement is made on each cycle in the record.

FALLTIME (Fall Time) is the time required for an edge to fall from the Top reference level (RT) to the Base reference level (RB). This measurement is made on each cycle in the record.

FREQuency is the reciprocal of Period. This measurement is made on each cycle in the record.
F2 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 2. This measurement is made across the entire record.

F4 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 4. This measurement is made across the entire record.

F8 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 8. This measurement is made across the entire record.

HIGH (Eye High) is the amplitude of a high (1) bit measured at a user specified location within the recovered unit interval. This measurement is made on each high bit in the record.

HEIGHT (Eye Height) is the minimum vertical eye opening at the center of the recovered unit interval. This measurement is made across the entire record.

HEIGHTBER (Eye Height@BER) is the predicted vertical eye opening that will be violated with a probability equal to the bit error rate. This measurement is made across the entire record.

HIGH

HIGHTIME (High Time) is the time the signal remains above the Top reference level (RT). This measurement is made on each cycle in the record.

HOLD (Hold Time) is the time between the specified Mid reference level crossing (RM) on the Clock source to the closest specified Mid reference level (RM) crossing on the Data source. This measurement is made on each specified Clock edge in the record.

JITTERSUMMARY (Jitter Summary) is a group consisting of the following measurements: TIE, TJ@BER, Eye Width@BER, Eye Height@BER, RJ-δδ, DJ-δδ, PJ, DDJ, DcD, F/2, F/4, F/8.

J2 is the total jitter at a bit error rate of 2.5e-3 (TJ@2.5e-3). This measurement is made across the entire record.

J9 is the total jitter at a bit error rate of 2.5e-10 (TJ@2.5e-10). This measurement is made across the entire record.

LOW (Eye Low) is the amplitude of a low (0) bit measured at a user specified location within the recovered unit interval. This measurement is made on each low bit in the record.

LOWTIME (Low Time) is the time the signal remains below the Base reference level (RB). This measurement is made on each cycle in the record.

MAXIMUM is the maximum data point. This measurement can be made across the entire record, or on each cycle in the record.

MEAN is the arithmetic mean of the data points. This measurement can be made across the entire record, or on each cycle in the record.

MINIMUM is the minimum data point. This measurement can be made across the entire record, or on each cycle in the record.
NDuty (Negative Duty Cycle) is the ratio of the Negative Pulse Width to the Period. This measurement is made on each cycle in the record.

\[ \text{Negative Duty Cycle} = \frac{\text{Negative Width}}{\text{Period}} \times 100\% \]

NPeriod (Duration N-Periods) is the time required to complete N cycles. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

NPJ (non-periodic jitter) is the portion of the BUJ (bounded uncorrelated jitter) that is random. BUJ excludes DDJ, DCD and RJ. This measurement is made across the entire record.

NOvershoot (Negative Overshoot) is the difference between Minimum and Base, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

\[ \text{Negative Overshoot} = \frac{\text{Base} - \text{Minimum}}{\text{Amplitude}} \times 100\% \]

NWIdth (Negative Pulse Width) is the time the signal remains below the Mid reference level (RM). This measurement is made on each cycle in the record.

PDUTY (Positive Duty Cycle) is the ratio of the Positive Pulse Width to the Period. This measurement is made on each cycle in the record.

\[ \text{Positive Duty Cycle} = \frac{\text{Positive Width}}{\text{Period}} \times 100\% \]

PERIOD is the time required to complete a cycle. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

PHASE is the ratio of the Skew between two sources to the Period of the first source. This measurement is made on each cycle in the record.

PHASENOISE (Phase Noise) is the RMS magnitude of all integrated jitter falling within a user specified offset range of the fundamental clock frequency. This measurement is made across the entire record.

PJ (periodic jitter) is the peak-to-peak amplitude of the uncorrelated sinusoidal components of the deterministic jitter. This measurement is made across the entire record.

PK2Pk (Peak-to-peak) is the difference between Maximum and Minimum. This measurement can be made across the entire record, or on each cycle in the record.

POvershoot (Positive Overshoot) is the difference between Maximum and Top, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

\[ \text{Positive Overshoot} = \frac{\text{Maximum} - \text{Top}}{\text{Amplitude}} \times 100\% \]

PWIDTH (Positive Pulse Width) is the time the signal remains above the Mid reference level (RM). This measurement is made on each cycle in the record.
QFACTOR (Q-Factor) is the ratio of the vertical eye opening to RMS vertical noise measured at a user specified location within the recovered unit interval. This measurement is made across the entire record.

RISE SLEWRATE (Rising Slew Rate) is the rate of change in voltage as an edge transitions from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RISE TIME Rise Time is the time required for an edge to rise from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

R J (random jitter) is the RMS magnitude of all random timing errors following a Gaussian distribution. This measurement is made across the entire record.

RJD I R AC (dual-dirac random jitter) is random jitter based on a simplifying assumption that the histogram of all deterministic jitter can be modeled as a pair of equal-magnitude Dirac functions. This measurement is made across the entire record.

RMS is the true Root Mean Square of the data points. This measurement can be made across the entire record, or on each cycle in the record.

SR J (sub-rate jitter) is the composite jitter due to periodic components at 1/2, 1/4 and 1/8 of the data rate. This measurement is made across the entire record.

SSCFREQDEV (SSC Frequency Deviation) is the spread spectrum clock frequency deviation. This measurement enables a time trend plot of the spread spectrum clock modulation profile. This measurement is made on each cycle in the record.

SSCMODRATE (SSC Modulation Rate) is the modulating frequency of a spread spectrum clock. This measurement is made on each cycle in the record.

SETUP (Setup Time) is the time between the specified Mid reference level (RM) crossing on the Data source to the closest specified Mid reference level (RM) crossing on the Clock source. This measurement is made on each specified Clock edge in the record.

SKEW Skew is the time between the specified Mid reference level (RM) crossing on one source to the following specified Mid reference level (RM) crossing on a second source. This measurement is made on each cycle in the record.

TIE (time interval error) is the difference, in time, between an edge in the source waveform and the corresponding edge in a recovered reference clock. This measurement is made on each edge in the waveform.

TIMEOUTSIDELEVEL Time Outside Level is the time the signal remains above the Top reference level (RT) and/or below the Base reference level (RB). This measurement is made on each occurrence in the record.

TJBER (total jitter at a specified bit error rate) is the predicted peak-to-peak amplitude of jitter that will only be exceeded with a probability equal to the bit error rate. This measurement is made across the entire record.
Commands listed in alphabetical order

TNTRATIO  
T/nT Ratio is the ratio of a non-transition bit voltage (2nd and subsequent bit voltage after a transition) to its nearest preceding transition bit voltage (1st bit voltage after the transition). Bit voltages are measured at the interpolated midpoint of the recovered unit interval. This measurement is made on each non-transition bit in the record.

TOP  
is the most common data value above the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

UNITINTERVAL  
(Unit Interval) is the time difference between two successive bits. This measurement is made on each bit in the record.

VDIFFXOVR  
(Differential Crossover) is the voltage level of a differential signal pair at the crossover points. This measurement is made at each crossover point in the record.

WIDTH  
(Eye Width) is the minimum horizontal eye opening at the user specified reference level. This measurement is made across the entire record.

WIDTHBER  
(Eye Width@BER) is the predicted horizontal eye opening that will be violated with a probability equal to the bit error rate. This measurement is made across the entire record.

Examples

MEASUREMENT:MEAS2:TYPE FREQUENCY defines measurement 2 as a measurement of the frequency of a waveform.

MEASUREMENT:MEAS1:TYPE? might return :MEASUREMENT:MEAS1:TYPE RMS, indicating that measurement 1 is defined to measure the RMS value of a waveform.

MEASUREMENT:MEAS<x>:WINDOWLENgth

This command sets or queries the window length for the measurement. The measurement number is specified by x.

Group  Measurement

Syntax  MEASUREMENT:MEAS<x>:WINDOWLENgth <NR3>
        MEASUREMENT:MEAS<x>:WINDOWLENgth?

Arguments  <NR3> is the value for the window length.

Examples  MEASUREMENT:MEAS1:WINDOWLENgth 10 sets the window length to 10.

MEASUrement:MINUI

This command sets or queries the minimum number of unit intervals required for BUJ analysis.

Group Measurement

Syntax

MEASUrement:MINUI <NR3>
MEASUrement:MINUI?

Arguments

<NR3> is the minimum number of unit intervals required for BUJ analysis.

Examples

MEASUREMENT:MINUI 1.0000E+6 sets the minimum number of unit intervals required for BUJ analysis to 1,000,000.

MEASUREMENT:MINUI? might return :MEASUREMENT:MINUI 1.0000E+6 indicating the minimum number of unit intervals required for BUJ analysis is 1,000,000.

MEASUrement:POPUlation:LIMIT:STATE

This command sets or queries the global population limit state for the measurement.

Group Measurement

Syntax

MEASUrement:POPUlation:LIMIT:STATE {OFF|ON|0|1 }
MEASUrement:POPUlation:LIMIT:STATE?

Arguments

OFF turns off the population limit.

ON turns on the population limit.

0 turns off the population limit.

1 turns on the population limit.

Examples

MEASUREMENT:POPULATION:LIMIT:STATE 1 turns on the population limit.

MEASUREment:POPULATION:LIMIT:VALUE

This command sets or queries the global population limit value for the measurement.

**Group**  
Measurement

**Syntax**  
MEASUREment:POPULATION:LIMIT:VALUE <NR1>  
MEASUREment:POPULATION:LIMIT:VALUE?

**Arguments**  
<NR1> is the current limit value.

**Examples**  


MEASUREment:REFLevels:ABSolute:FALLHigh

This command sets or queries the value used as the high reference level of the falling edge when the measurement's reflevel method is set to absolute.

**Group**  
Measurement

**Syntax**  
MEASUREment:REFLevels:ABSolute:FALLHigh <NR3>  
MEASUREment:REFLevels:ABSolute:FALLHigh?

**Arguments**  
<NR3> is the value used as the high reference level of the falling edge when the measurement's reflevel method is set to absolute.

**Examples**  
MEASUREMENT:REFLevels:ABSolute:FALLHigh 1.5 sets the high reference level to 1.5 V.

MEASUREMENT:REFLevels:ABSolute:FALLHigh? might return :MEASUREMENT:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000 indicating the high reference level is 1.0 V.
**MEASUrement:REFLevels:ABSolute:FALLLow**

This command sets or queries the value used as the low reference level of the falling edge when the measurement’s ref level method is set to absolute.

**Group**  
Measurement

**Syntax**  
MEASUrement:REFLevels:ABSolute:FALLLow <NR3>  
MEASUrement:REFLevels:ABSolute:FALLLow?

**Arguments**  
<NR3> is the value used as the low reference level of the falling edge.

**Examples**  
MEASUrement:REFLevels:ABSolute:FALLLow -1.5 sets the low reference level to -1.5 V.  
MEASUrement:REFLevels:ABSolute:FALLLow? might return :MEASUREMENT:REFLEVELS:ABSOLUTE:FALLLOW -1.0000 indicating the low reference level is -1.0 V.

**MEASUrement:REFLevels:ABSolute:FALLMid**

This command sets or queries the value used as the mid reference level of the falling edge when the measurement’s ref level method is set to absolute.

**Group**  
Measurement

**Syntax**  
MEASUrement:REFLevels:ABSolute:FALLMid <NR3>  
MEASUrement:REFLevels:ABSolute:FALLMid?

**Arguments**  
<NR3> is the value used as the mid reference level of the falling edge.

**Examples**  
MEASUrement:REFLevels:ABSolute:FALLMid 10.0E-3 sets the mid reference level to 10.0 mV.  
MEASUrement:REFLevels:ABSolute:FALLMid? might return :MEASUREMENT:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0 indicating the mid reference level is 0.0 V.
MEASUrement:REFLevels:ABSolute:HYSTeresis

This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.

Group: Measurement

Syntax:

- MEASUrement:REFLevels:ABSolute:HYSTeresis <NR3>
- MEASUrement:REFLevels:ABSolute:HYSTeresis?

Arguments:

- `<NR3>` is the value of the hysteresis of the reference level.

Examples:

- MEASUrement:REFLevels:ABSolute:HYSTeresis 20.0E-3 sets the hysteresis to 20.0 mV.
- MEASUrement:REFLevels:ABSolute:HYSTeresis? might return :MEASUREMENT:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3 indicating the hysteresis is set to 30.0 mV.

MEASUrement:REFLevels:ABSolute:RISEHigh

This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.

Group: Measurement

Syntax:

- MEASUrement:REFLevels:ABSolute:RISEHigh <NR3>
- MEASUrement:REFLevels:ABSolute:RISEHigh?

Arguments:

- `<NR3>` is the value used as the high reference level of the rising edge.

Examples:

- MEASUrement:REFLevels:ABSolute:RISEHigh 1.5 sets the high reference to 1.5 V.
- MEASUrement:REFLevels:ABSolute:RISEHigh? might return :MEASUREMENT:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000 indicating the high reference is 1.0 V.
**MEASUrement:REFLevels:ABSolute:RISELow**

This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.

**Group**  
Measurement

**Syntax**  
MEASUrement:REFLevels:ABSolute:RISELow <NR3>  
MEASUrement:REFLevels:ABSolute:RISELow?

**Arguments**  
<NR3> is the value used as the the low reference level of the rising edge

**Examples**  
MEASUrement:REFLevels:ABSolute:RISELow -1.5 sets the low reference level to -1.5 V.  
MEASUrement:REFLevels:ABSolute:RISELow? might return :MEASUREMENT:REFLEVELS:ABSOLUTE:RISELOW -1.0000 indicating the low reference level is -1.0 V.

**MEASUrement:REFLevels:ABSolute:RISEMid**

This command sets or queries the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute.

**Group**  
Measurement

**Syntax**  
MEASUrement:REFLevels:ABSolute:RISEMid <NR3>  
MEASUrement:REFLevels:ABSolute:RISEMid?

**Arguments**  
<NR3> is the mid reference level of the rising edge.

**Examples**  
MEASUrement:REFLevels:ABSolute:RISEMid 10.0E-3 sets the mid reference to 10.0 mV.  
MEASUrement:REFLevels:ABSolute:RISEMid? might return :MEASUREMENT:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0 indicating the mid reference is 0.0 V.

**MEASUrement:REFLevels:ABSolute:TYPE**

This command sets or queries the reference level type for the measurement.
**MEASUrement:REFLevels:ABSolute:TYPE**

Group: Measurement

Syntax: MEASUrement:REFLevels:ABSolute:TYPE \{SAME\|UNIQue\}

Syntax: MEASUrement:REFLevels:ABSolute:TYPE?

Arguments:

- SAME specifies that the absolute levels are set the same.
- UNIQue specifies that the absolute levels can be set independently.

Examples:

- MEASUrement:REFLevels:ABSolute:TYPE UNIQue specifies that the absolute levels can be set independently.

**MEASUrement:REFLevels:BASETop**

This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement.

Group: Measurement

Syntax: MEASUrement:REFLevels:BASETop \{AUTO\|MINMax\|MEANhistogram\|MODEhistogram\|EYEhistogram\}

Syntax: MEASUrement:REFLevels:BASETop?

Arguments: Arguments are the base top methods.

Examples:

- MEASUREMENT:REFLEVELS:BASETOP MINMAX selects the MINMAX base top method.
- MEASUREMENT:REFLEVELS:BASETOP? might return :MEASUREMENT:REFLEVELS:BASETOP MINMAX indicating the base top method is MINMAX.

**MEASUrement:REFLevels:JITTERMODE**

This command sets or queries how often reference levels are calculated on Jitter measurements. If the mode is set to Latch, ref levels are calculated only on the first acquisition after a statistics reset. If it is set to Continuous, reference levels are calculated on every acquisition.
Commands listed in alphabetical order

**MEASUREMENT:REFLEVELS:JITTERMODE**

**Group** Measurement

**Syntax**

```
MEASUREMENT:REFLEVELS:JITTERMODE {CONTinuous|LATch}
MEASUREMENT:REFLEVELS:JITTERMODE?
```

**Arguments**

- CONTinuous specifies that reference levels are calculated on every acquisition.
- LATch specifies that reference levels are calculated only on the first acquisition after a statistics reset.

**Examples**

- `MEASUREMENT:REFLEVELS:JITTERMODE CONTINUOUS` specifies that reference levels are calculated on every acquisition.
- `MEASUREMENT:REFLEVELS:JITTERMODE` might return LATch indicating that reference levels are calculated only on the first acquisition after a statistics reset.

**MEASUREMENT:REFLEVELS:METHod**

This command sets or queries the method used to calculate reference levels for the measurement.

**Group** Measurement

**Syntax**

```
MEASUREMENT:REFLEVELS:METHod {PERCent|ABSolute}
MEASUREMENT:REFLEVELS:METHod?
```

**Arguments**

- PERCent specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the `MEASUREMENT:REFLEVELS:REFLEVEL:PERCENT` commands.
- ABSolute specifies that the reference levels are set explicitly using the `MEASUREMENT:REFLEVELS:REFLEVEL:ABSOLUTE` commands. This method is useful when precise values are required.

**Examples**

- `MEASUREMENT:REFLEVELS:METHod ABSOLUTE` specifies that the reference levels are set explicitly.
- `MEASUREMENT:REFLEVELS:METHod?` might return `:MEASUREMENT:REFLEVELS:METHod PERCENT` indicating the reference levels are calculated as a percent relative to HIGH and LOW.
**MEASUrement:REFLevels:MODE**

This command sets or queries how often reference levels are calculated.

**Group** Measurement

**Syntax**

```
MEASUrement:REFLeveles:MODE {LATCH|CONTinuous}  
MEASUrement:REFLeveles:MODE?
```

**Arguments**

- **LATCH** calculates reference levels only on the first acquisition after a statistics reset.
- **CONTinuous** calculates reference levels on every acquisition.

**Examples**

```
MEASUREMENT:REFLEVELS:MODE CONTINUOUS calculates reference levels on every acquisition.
MEASUREMENT:REFLEVELS:MODE? might return :MEASUREMENT:REFLEVELS:MODE LATCH indicating reference levels are calculated only on the first acquisition after a statistics reset.
```

**MEASUrement:REFLevels:PERCent:FALLHigh**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent.

**Group** Measurement

**Syntax**

```
MEASUrement:REFLeveles:PERCent:FALLHigh <NR3>  
MEASUrement:REFLeveles:PERCent:FALLHigh?
```

**Arguments**

- `<NR3>` is the percentage used to calculate the high reference level of the falling edge.

**Examples**

```
MEASUREMENT:REFLEVELS:PERCENT:FALLHIGH 90 sets the high reference level to 90%.
MEASUREMENT:REFLEVELS:PERCENT:FALLHigh? might return :MEASUREMENT:REFLEVELS:PERCENT:FALLHIGH 80.0000 indicating the high reference level is 80%.
```
MEASUrement:REFLev els:PERCent:FALLow

This command sets or queries the percentage (where 100% is equal to TOP and 
0% is equal to BASE) used to calculate the mid reference level of the falling edge 
when the measurement’s ref level method is set to percent.

Group       Measurement

Syntax      MEASUrement:REFLev els:PERCent:FALLow <NR3>
            MEASUrement:REFLev els:PERCent:FALLow?

Arguments   <NR3> is the percentage used to calculate the mid reference level of the falling edge.

Examples    MEASUrement:REFLev els:PERCent:FALLow 10 sets the low reference level to 10%.
            MEASUrement:REFLev els:PERCent:FALLow? might return
            :MEASUREMENT:REFLEVELS:PERCENT:FALLow 20.0000 indicating the low reference level is 20%.

MEASUrement:REFLev els:PERCent:FALLMid

This command sets or queries the percentage (where 100% is equal to TOP and 
0% is equal to BASE) used to calculate the mid reference level of the falling edge 
when the measurement’s ref level method is set to percent.

Group       Measurement

Syntax      MEASUrement:REFLev els:PERCent:FALLMid <NR3>
            MEASUrement:REFLev els:PERCent:FALLMid?

Arguments   <NR3> is the percentage used to calculate the mid reference level of the falling edge.

Examples    MEASUrement:REFLev els:PERCent:FALLMid 55 sets the mid reference level to 55%.
            MEASUrement:REFLev els:PERCent:FALLMid? might return
            :MEASUREMENT:REFLEVELS:PERCENT:FALLMID 50.0000 indicating the mid reference level is 50%.
**MEASUrement:REFLevels:PERCent:HYSTeresis**

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent.

**Group**
Measurement

**Syntax**
- `MEASUREMENT:REFLeveLS:PERCent:HYSTeresis <NR3>`
- `MEASUREMENT:REFLeveLS:PERCent:HYSTeresis?`

**Arguments**
- `<NR3>` is the percentage used to calculate the hysteresis of the reference level.

**Examples**
- `MEASUREMENT:REFLeveLS:PERCent:HYSTeresis 3` sets the hysteresis to 3%.
- `MEASUREMENT:REFLeveLS:PERCent:HYSTeresis?` might return `:MEASUREMENT:REFLEVELS:PERCENT:HYSTERESIS 5.0000` indicating the hysteresis is set to 5%.

---

**MEASUrement:REFLeveLS:PERCent:RISEHigh**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent.

**Group**
Measurement

**Syntax**
- `MEASUREMENT:REFLeveLS:PERCent:RISEHigh <NR3>`
- `MEASUREMENT:REFLeveLS:PERCent:RISEHigh?`

**Arguments**
- `<NR3>` is the percentage used to calculate the high reference level of the rising edge.

**Examples**
- `MEASUREMENT:REFLeveLS:PERCent:RISEHigh 90` sets the high reference level to 90%.
- `MEASUREMENT:REFLeveLS:PERCent:RISEHigh?` might return `:MEASUREMENT:REFLEVELS:PERCENT:RISEHIGH 80.0000` indicating the high ref level is 80%.
**MEASUrement:REFLevels:PERCent:RISELow**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent.

**Group**  Measurement

**Syntax**

- `MEASUREMENT:REFLevels:PERCent:RISELow <NR3>`
- `MEASUREMENT:REFLevels:PERCent:RISELow?`

**Arguments**

- `<NR3>` is the percentage used to calculate the low reference level of the rising edge.

**Examples**

- `MEASUREMENT:REFLevels:PERCent:RISELow 10` sets the low reference level to 10%.

**MEASUrement:REFLevels:PERCent:RISEMid**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent.

**Group**  Measurement

**Syntax**

- `MEASUREMENT:REFLevels:PERCent:RISEMid <NR3>`
- `MEASUREMENT:REFLevels:PERCent:RISEMid?`

**Arguments**

- `<NR3>` is the percentage used to calculate the mid reference level of the rising edge.

**Examples**

- `MEASUREMENT:REFLevels:PERCent:RISEMid 55` sets the mid reference level to 55%.
- `MEASUREMENT:REFLevels:PERCent:RISEMid?` might return `:MEASUREMENT:REFLEVELS:PERCENT:RISEMIID 50.0000` indicating the mid reference level is 50%.
MEASUrement:REFLevels:PERCent:TYPE

This command sets or queries the reference level percent type for the measurement.

Group  Measurement

Syntax  MEASUrement:REFLevels:PERCent:TYPE  
{TENNinety|TWENtyeighty|CUSTom}
MEASUrement:REFLevels:PERCent:TYPE?

Arguments

TENNinety sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.
TWENtyeighty sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.
CUSTom allows setting other reference level percents.

Examples

MEASUrement:REFLevels:PERCent:TYPE TENNINETY sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.
MEASUrement:REFLevels:PERCent:TYPE? might return
:MEASUREMENT:REFLEVELS:PERCENT:TYPE CUSTOM indicating that custom reference levels can be set.

MEASUrement:REFLevels:TYPE

This command sets or queries the shared reference level method used for sources of measurement calculations.

Group  Measurement

Syntax  MEASUrement:REFLevels:TYPE  
{GLOBal|PERSource}
MEASUrement:REFLevels:TYPE?

Arguments

GLOBal shares reference levels across measurements.
PERSource causes reference levels to be used on individual measurements.

Examples

MEASUREMENT:REFLEVELS:TYPE PERSource causes reference levels to be used on individual measurements.
MEASUrement:REF<x>:REFLevels:ABSolute:FALLHigh

This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.

Group  Measurement

Syntax  MEASUrement:REF<x>:REFLevels:ABSolute:FALLHigh <NR3>
        MEASUrement:REF<x>:REFLevels:ABSolute:FALLHigh?

Arguments  <NR3> is the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.

Examples  MEASUrement:REF1:REFLevels:ABSolute:FALLHigh 1.5 sets the reference level to 1.5 V.
          MEASUrement:REF1:REFLevels:ABSolute:FALLHigh? might return 
          :MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000 indicating
          the reference level is 1.0 V.

MEASUrement:REF<x>:REFLevels:ABSolute:FALLLow

This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.

Group  Measurement

Syntax  MEASUrement:REF<x>:REFLevels:ABSolute:FALLLow <NR3>
        MEASUrement:REF<x>:REFLevels:ABSolute:FALLLow?

Arguments  <NR3> is the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.

Examples  MEASUrement:REF1:REFLevels:ABSolute:FALLLow -1.5 sets the low reference level to -1.5 V.
MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLLow?
might return:
:MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLLOW -1.0000 indicating
the low reference level is -1.0 V.

MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:FALLMid
This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.

Group Measurement
Syntax MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:FALLMid <NR3>
MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:FALLMid?
Arguments <NR3> is the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.
Examples MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLMid 50.0E-3 sets the mid reference level to 50.0 mV.
MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLMid? might return:
:MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0 indicating the reference level is 0.0 V.

MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:HYSTeresis
This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.

Group Measurement
Syntax MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:HYSTeresis <NR3>
MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:HYSTeresis?
Arguments <NR3> is the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.
Examples MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:HYSTeresis 20.0E-3 sets the hysteresis to 20.0 mV.
MEASurement:REF<x>:REFLevels:ABSolute:HYSTeresis?

This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.

**Group** Measurement

**Syntax**

MEASurement:REF<x>:REFLevels:ABSolute:RISEHigh <NR3>

MEASurement:REF<x>:REFLevels:ABSolute:RISEHigh?

**Arguments**

<NR3> is the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.

**Examples**

MEASurement:REF1:REFLevels:ABSolute:RISEHigh 1.5 sets the reference level to 1.5 V.

MEASurement:REF1:REFLevels:ABSolute:RISEHigh? might return :

:MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000 indicating the reference level is 1.0 V.

MEASUrement:REF<x>:REFLevels:ABSolute:RISELow

This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.

**Group** Measurement

**Syntax**

MEASurement:REF<x>:REFLevels:ABSolute:RISELow <NR3>

MEASurement:REF<x>:REFLevels:ABSolute:RISELow?

**Arguments**

<NR3> is the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.

**Examples**

MEASurement:REF1:REFLevels:ABSolute:RISELow -1.5 sets the reference level to -1.5 V.

MEASurement:REF1:REFLevels:ABSolute:HYSTeresis? might return :

:MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3 indicating the hysteresis is 30.0 mV.
MEASUREMENT:REF1:REF_LEVELS:ABSOLUTE:RISELOW? might return
:MEASUREMENT:REF1:REF_LEVELS:ABSOLUTE:RISELOW -1.0000 indicating
the reference level is -1.0 V.

MEASUREMENT:REF<x>:REF_LEVELS:ABSOLUTE:RISE Mid
This command sets or queries the value used as the mid reference level of the
rising edge when the measurement's ref level method is set to absolute.

Group Measurement
Syntax
MEASUREMENT:REF<x>:REF_LEVELS:ABSOLUTE:RISE Mid <NR3>
MEASUREMENT:REF<x>:REF_LEVELS:ABSOLUTE:RISE Mid?
Arguments
<NR3> is the value used as the mid reference level of the rising edge when the
measurement's ref level method is set to absolute.
Examples
MEASUREMENT:REF1:REF_LEVELS:ABSOLUTE:RISE Mid 10.0E-3 sets the
reference level to 10.0 mV.
MEASUREMENT:REF1:REF_LEVELS:ABSOLUTE:RISE Mid? might return
:MEASUREMENT:REF1:REF_LEVELS:ABSOLUTE:RISE Mid 0.0E+0 indicating
the mid reference level is 0.0 V.

MEASUREMENT:REF<x>:REF_LEVELS:ABSOLUTE:TYPE
This command sets or queries the reference level type for the measurement.

Group Measurement
Syntax
MEASUREMENT:REF<x>:REF_LEVELS:ABSOLUTE:TYPE {SAME|UNIQUE}
MEASUREMENT:REF<x>:REF_LEVELS:ABSOLUTE:TYPE?
Arguments
SAME specifies that the absolute levels are set the same.
UNIQUE specifies that the absolute levels can be set independently.
Examples
MEASUREMENT:REF1:REF_LEVELS:ABSOLUTE:TYPE UNIQUE specifies that the
absolute levels can be set independently.
MEASUrement:REF<x>:REFLevels:BASETop

This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement.

Group  Measurement
Syntax   MEASUrement:REF<x>:REFLevels:BASETop

Arguments
AUTO automatically chooses a reference level method.
MINMax specifies that reference levels are relative to the measurement MIN and MAX.
MEANhistogram specifies that reference levels are relative to the histogram mean BASE and TOP.
MODEhistogram specifies that reference levels are relative to the histogram mode BASE and TOP.
EYEHistogram specifies that reference levels are relative to the eye histogram BASE and TOP.

Examples
MEASUrement:REF1:REFLevels:BASETop MINMAX specifies that reference levels are relative to the measurement MIN and MAX.
MEASUrement:REF1:REFLevels:BASETop? might return
:MEASUREMENT:REF1:REFLEVELS:BASETOP AUTO indicating the instrument automatically chooses a reference level method.

MEASUrement:REF<x>:REFLevels:METHod

This command sets or queries the method used to calculate reference levels for the measurement.

Group  Measurement
Syntax   MEASUrement:REF<x>:REFLevels:METHod {PERCent|ABSolute}
MEASUrement:REF<x>:REFLevels:METHod?
Arguments

PERCent specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUrement:REF<x>:REFlevel:PERCent commands.

ABSolute specifies that the reference levels are set explicitly using the MEASUrement:REF<x>:REFlevel:ABSolute commands. This method is useful when precise values are required.

Examples

MEASUrement:REF1:REFLevels:METHOD ABSOLUTE specifies that the reference levels are set explicitly.

MEASUrement:REF1:REFLevels:METHOD? might return :MEASUREMENT:REF1:REFLEVELS:METHOD PERCENT indicating the reference levels are calculated as a percent relative to HIGH and LOW.

**MEASUrement:REF<x>:REFLevels:PERCent:FALLHigh**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax

MEASUrement:REF<x>:REFLevels:PERCent:FALLHigh <NR3>

MEASUrement:REF<x>:REFLevels:PERCent:FALLHigh?

Arguments

<NR3> is the percentage used to calculate the high reference level of the falling edge

Examples

MEASUrement:REF1:REFLevels:PERCent:FALLHigh 95 sets the reference level to 95%.

MEASUrement:REF1:REFLevels:PERCent:FALLHigh? might return :MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLHIGH 90.0000 indicating the reference level is 90%.

**MEASUrement:REF<x>:REFLevels:PERCent:FALLLow**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement's ref level method is set to percent.
Commands listed in alphabetical order

**MEASUrement:REF<x>:REFLevels:PERCent:FALLLow**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement's ref level method is set to percent.

**Group** Measurement

**Syntax**

- `MEASUrement:REF<x>:REFLevels:PERCent:FALLLow <NR3>`
- `MEASUrement:REF<x>:REFLevels:PERCent:FALLLow?`

**Arguments**

- `<NR3>` is the percentage used to calculate the low reference level.

**Examples**

- `MEASUrement:REF1:REFLevels:PERCent:FALLLow 5` sets the low reference level to 5%.

**MEASUrement:REF<x>:REFLevels:PERCent:FALLMid**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent.

**Group** Measurement

**Syntax**

- `MEASUrement:REF<x>:REFLevels:PERCent:FALLMid <NR3>`
- `MEASUrement:REF<x>:REFLevels:PERCent:FALLMid?`

**Arguments**

- `<NR3>` is the percentage used to calculate the mid reference level of the falling edge.

**Examples**

- `MEASUrement:REF1:REFLevels:PERCent:FALLMid 55` sets the reference level to 50%.
- `MEASUrement:REF1:REFLevels:PERCent:FALLMid?` might return `:MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLMID 50.0000` indicating the mid reference level is 50%.

**MEASUrement:REF<x>:REFLevels:PERCent:HYSTeresis**

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent.
Commands listed in alphabetical order

**Group** Measurement

**Syntax**

MEASUrement:REF<x>:REFLevels:PERCent:HYSTeresis  <NR3>  
MEASUrement:REF<x>:REFLevels:PERCent:HYSTeresis?

**Arguments**

<NR3> is the percentage used to calculate the hysteresis of the reference level.

**Examples**

MEASUrement:REF1:REFLevels:PERCent:HYSTeresis 2 sets the hysteresis to 2%.

MEASUrement:REF1:REFLevels:PERCent:HYSTeresis? might return 
:MEASUREMENT:REF1:REFLEVELS:PERCENT:HYSTERESIS 5.0000 indicating the hysteresis is 5%.

**MEASUrement:REF<x>:REFLevels:PERCent:RISEHigh**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent.

**Group** Measurement

**Syntax**

MEASUrement:REF<x>:REFLevels:PERCent:RISEHigh  <NR3>  
MEASUrement:REF<x>:REFLevels:PERCent:RISEHigh?

**Arguments**

<NR3> is the percentage used to calculate the high reference level of the rising edge.

**Examples**

MEASUrement:REF1:REFLevels:PERCent:RISEHigh 95 sets the reference level to 95%.

MEASUrement:REF1:REFLevels:PERCent:RISEHigh? might return 
:MEASUREMENT:REF1:REFLEVELS:PERCENT:RISEHIGH 90.0000 indicating the reference level is 90%.

**MEASUrement:REF<x>:REFLevels:PERCent:RISELow**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent.
Commands listed in alphabetical order

**MEASUrement:REF<x>:REFLevels:PERCent:RISELow**

**Group**  Measurement

**Syntax**  
MEASUrement:REF<x>:REFLevels:PERCent:RISELow <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:RISELow?

**Arguments**  
<NR3> is the percentage used to calculate the low reference level of the rising edge.

**Examples**  
MEASUrement:REF1:REFLevels:PERCent:RISELow 5 sets the low reference level to 5%.

**MEASUrement:REF<x>:REFLevels:PERCent:RISEMId**

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent.

**Group**  Measurement

**Syntax**  
MEASUrement:REF<x>:REFLevels:PERCent:RISEMId <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:RISEMId?

**Arguments**  
<NR3> is the percentage used to calculate the mid reference level of the rising edge.

**Examples**  
MEASUrement:REF1:REFLevels:PERCent:RISEMId 55 sets the mid reference level to 55%.
MEASUrement:REF1:REFLevels:PERCent:RISEMId? might return :MEASUREMENT:REF1:REFLEVELS:PERCENT:RISEMID 50.0000 indicating the mid reference level is 50%.

**MEASUrement:REF<x>:REFLevels:PERCent:TYPE**

This command sets or queries the reference level percent type for the measurement.

**Group**  Measurement
Syntax

MEASUrement:REF<x>:REFLeve<ls>:PERCent:TYPE
{TENNinety|TWENtyeighty|CUSTom}
MEASUrement:REF<x>:REFLeve<ls>:PERCent:TYPE?

Arguments

TENNinety sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

TWENtyeighty sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

CUSTom allows setting other reference level percents.

Examples

MEASUrement:REF1:REFLeve<ls>:PERCent:TYPE TWENTYEIGHTY sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

MEASUrement:REF1:REFLeve<ls>:PERCent:TYPE? might return :MEASUREMENT:REF1:REFLEVELS:PERCENT:TYPE TENNINETY indicating the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

**MEASUrement:STATIstics:CYCLEMode**

This command turns on and off cycle to cycle measurement statistics tracking and affects computation and display of cycle-cycle statistics in the Measurement Result table. It affects measurement statistics after being enabled and after new data is acquired and measured.

Group

Measurement

Syntax

MEASUrement:STATIstics:CYCLEMode {OFF|ON|0|1}
MEASUrement:STATIstics:CYCLEMode?

Arguments

OFF turns off statistics for all measurements. This is the default value.

ON turns on statistics and displays all statistics for each measurement.

0 turns off statistics for all measurements.

1 turns on statistics and displays all statistics for each measurement.

Examples

MEASUREMENT:STATISTICS:CYCLEMODE OFF turns off statistics for all measurements.

MEASUREMENT:STATISTICS:CYCLEMODE? might return :MEASUREMENT:STATISTICS:CYCLEMODE 1 indicating that statistics are displayed for each measurement.
NEWpass (No Query Form)

This command (no query form) changes the password that enables access to password protected data. The PASSWord command must be successfully executed before using this command or an execution error will be generated.

Group     Miscellaneous

Syntax     NEWpass <QString>

Related Commands
PASSWord
*PUD

Arguments
QString is the new password, which can contain up to 10 characters.

Examples
NEWPASS"mypassword" creates a new password (mypassword) for accessing your protected data.

*OPC

This command generates the operation complete message in the Standard Event Status Register (SESR) when all pending commands that generate an OPC message are complete. The *OPC? query places the ASCII character “1” into the output queue when all such OPC commands are complete. The *OPC? response is not available to read until all pending operations finish. For a complete discussion of the use of these registers and the output queue, see Registers and Queues.

The *OPC command allows you to synchronize the operation of the instrument with your application program. For more information, see Synchronization Methods.

Table 2-46: Commands that Generate an OPC Message

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQquire:STATE ON or</td>
</tr>
<tr>
<td>ACQquire:STATE RUN or</td>
</tr>
<tr>
<td>ACQquire:STATE 1 (when ACQquire:STOPAfter is set to SEQuence)</td>
</tr>
<tr>
<td>AUTOSet, CALibrate:INTERNal, CALibrate:INTERNal:STARt</td>
</tr>
<tr>
<td>Refer to the Service Manual.</td>
</tr>
<tr>
<td>DIAg:STATE, SAVe:IMAGe, SAVe:WAVEform, CH&lt;x&gt;:PRObe:AUTOZero, CH&lt;x&gt;:PRObe:DEGAUSS</td>
</tr>
<tr>
<td>FACTory, RECALL:SETUp, *RST, TEKSecure</td>
</tr>
</tbody>
</table>
Commands listed in alphabetical order

**Group** Status and Error

**Syntax**

*OPC  
*OPC?

**Related Commands**

BUSY?  
*WAI

**Examples**

*OPC generates the operation complete message in the SESR at the completion of all pending OPC operations.  
*OPC? might return 1 to indicate that all pending OPC operations are finished.

***OPT? (Query Only)**

This query-only command returns a comma separated list of installed options as an arbitrary ASCII string (no quotes) of the form:

<optionCode>:<optionDescription>,<optionCode>:<optionDescription>...

The last section of each entry (the text following the last hyphen) indicates the license type.

If no options are found, NONE is returned.

**Group** Status and Error

**Syntax**

*OPT?

**Examples**

*OPT? response (with each option listed on a separate line for clarity):

5-BW-1000 – 1 GHz,  
5-DJA – Advanced Jitter and Eye Analysis – NODE.

**PASSWord (No Query Form)**

This command (no query form) enables the *PUD and NEWpass set commands. Sending PASSWord without any arguments disables these same commands. Once the password is successfully entered, the *PUD and NEWpass commands are enabled until the instrument is powered off, or until the FACTory command, the PASSWord command with no arguments, or the *RST command is issued.
To change the password, you must first enter the valid password with the PASSWORD command and then change to your new password with the NEWpass command. Remember that the password is case sensitive.

**GROUP** Miscellaneous

**Syntax**

```
PASSWORD <QString>
```

**Related Commands**

NEWpass

*PUD

**Arguments**

<QString> is the password, which can contain up to 10 characters. The factory default password is "XYZZY" and is always valid.

**Examples**

PASSWORD "XYZZY" enables the *PUD and NEWPass set commands.

PASSWORD disables the *PUD and NEWPass set commands. You can still use the query version of *PUD.

**PAUSE (No Query Form)**

This command causes the interface to pause the specified number of seconds before processing any other commands.

**GROUP** Miscellaneous

**Syntax**

```
PAUSE <NR3>
```

**Arguments**

<NR3> is the specified number of seconds the interface is to pause before processing any other commands. The pause time is specified as a floating point value in seconds and must be > 0.0 and ≥1800.0.

**Examples**

PAUSE 10.0e0;:ACQUIRE:NUMACQ causes the interface to pause 10 seconds before returning the number of acquisitions.

**PLOT:ADDNew (No Query Form)**

This command adds the specified plot.
Group  Plot

Syntax  PLOT:ADDNew <QString>

Arguments  <QString> is the specified plot. The argument is of the form “PLOT<NR1>”, where <NR1> ≥ 1.

Examples  PLOT:ADDNEW "PLOT1" adds PLOT1.

PLOT:DELETE (No Query Form)

This command deletes the specified plot.

Group  Plot

Syntax  PLOT:DELETE <QString>

Arguments  <QString> is the specified plot. Argument is of the form “PLOT<NR1>”, where <NR1> is ≥ 1.

Examples  PLOT:DELETE "PLOT1" deletes PLOT1.

PLOT:LIST? (Query Only)

This command lists all currently defined plots.

Group  Plot

Syntax  PLOT:LIST?

Returns  A list of all currently defined plots is returned.

Examples  PLOT:LIST? might return :PLOT:LIST PLOT1, PLOT3, PLOT4, PLOT5, PLOT6, PLOT7 listing all currently defined plots.
PLOT:PLOT<x>:BATHTub:BER

This command sets or queries the bathtub BER value.

Group  Plot

Syntax  PLOT:PLOT<x>:BATHTub:BER  <NR1>
        PLOT:PLOT<x>:BATHTub:BER?

Arguments  <NR1> is the bathtub BER value.

Examples  PLOT:PLOT1:BATHTub:BER  16 sets the BER value to 16.

PLOT:PLOT<x>:BATHTub:XAXISUnits

This command sets or queries the X-Axis unit, either unit intervals or seconds.

Group  Plot

Syntax  PLOT:PLOT<x>:BATHTub:XAXISUnits  {UNITIntervals|SECOnds}
        PLOT:PLOT<x>:BATHTub:XAXISUnits?

Arguments  UNITIntervals specifies units as unit intervals.
            SECOnds specifies units as seconds.

Examples  PLOT:PLOT1:BATHTub:XAXISUnits  SECONDS sets the units to seconds.
          PLOT:PLOT1:BATHTub:XAXISUnits?  might return SECONDS.

PLOT:PLOT<x>:BITType

This command sets or queries the bit type to display for the specified eye diagram plot.

Group  Plot
Syntax
PLOT:PLOT<x>:BITType {ALLBits|TRANSition|NONTRANSition}
PLOT:PLOT<x>:BITType?

Arguments
PLOT<x> is the plot number.

ALLBits sets the eye diagram plot to show both transition and nontransition bits.
TRANSition sets the eye diagram plot to show only bits where a logic level transition occurs.
NONTRANSition sets the eye diagram plot to show only bits where no logic level transition occurs.

Examples
PLOT:PLOT1:BITType TRANSition sets the eye diagram in Plot 1 to only show transition bits.
PLOT:PLOT3:BITType? might return ALLBITS, indicating that the eye diagram in Plot 3 is set to show both transition and nontransition bits in the plot.

PLOT:PLOT<x>:LABel:COLor

This command sets or queries the color of the specified trend label. This command/query only applies to Time Trend plots.

Group Plot

Syntax
PLOT:PLOT<x>:LABel:COLor <QString>

Arguments
<QString> is the label color. The default color is specified by a quoted empty string, and is the only available color.

Examples
PLOT:PLOT2:LABel:COLor " " sets the plot color to the default color.
PLOT:PLOT2:LABel:COLor? might return :PLOT:PLOT2:LABel:COLor " " indicating the plot color is the default color.

PLOT:PLOT<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified trend label. This command/query only applies to Time Trend plots.

Group Plot
Syntax  PLOT:PLOT<x>:LABel:FONT:BOLD {<NR1>|OFF|ON}

Arguments  <NR1> = 0 disables bold font; any other value turns this feature on.
            OFF disables bold font.
            ON enables bold font.

Examples  PLOT:PLOT1:LABEL:FONT:BOLD 1 sets the label to a bold font.
                :PLOT:PLOT1:LABEL:FONT:BOLD 1 indicating the label is bold.

PLOT:PLOT<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the specified trend label. This command/query only applies to Time Trend plots.

Group  Plot

Syntax  PLOT:PLOT<x>:LABEL:FONT:ITALIC {<NR1>|OFF|ON}

Arguments  <NR1> = 0 disables italic font; any other value turns this feature on.
            OFF disables italic font.
            ON enables italic font.

Examples  PLOT:PLOT1:LABEL:FONT:ITALIC 1 sets the font to italic.
           PLOT:PLOT1:LABEL:FONT:ITALIC? might return
                :PLOT:PLOT1:LABEL:FONT:ITALIC 0 indicating the font is not italic.

PLOT:PLOT<x>:LABEL:FONT:SIZE

This command sets or queries the font size of the specified trend label. This command/query only applies to Time Trend plots.

Group  Plot

Syntax  PLOT:PLOT<x>:LABEL:FONT:SIZE <NR1>
Arguments  

<NR1> is the font size.

Examples  

PLOT:PLOT1:LABel:FONT:SIZE 12 sets the font size to 12 points.

PLOT:PLOT1:LABel:FONT:SIZE? might return 

:PLOT:PLOT1:LABEL:FONT:SIZE 72 indicating the font size is 72 points.

**PLOT:PLOT<x>:LABel:FONT:TYPE**

This command sets or queries the font type of the specified trend label, such as Arial or Times New Roman. This command/query only applies to Time Trend plots.

Group  Plot

Syntax  

PLOT:PLOT<x>:LABel:FONT:TYPE <QString>

Arguments  

<QString> is the font type: Times New Roman, Arial, Frutiger LT Std 55 Roman, DejaVu Sans, DejaVu Sans Mono, Frutiger LT Std, Monospace, Sans Serif, Serif, Ubuntu, Ubuntu Condensed, and Ubuntu Mono.

Examples  

PLOT:PLOT1:LABel:FONT:TYPE Arial sets the font type to Arial.

PLOT:PLOT1:LABel:FONT:TYPE? might return 

:PLOT:PLOT1:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman" indicating the font type is Frutiger LT Std 55 Roman.

**PLOT:PLOT<x>:LABel:FONT:UNDERline**

This command sets or queries the underline state of the specified trend label. This command/query only applies to Time Trend plots.

Group  Plot

Syntax  

PLOT:PLOT<x>:LABel:FONT:UNDERline {<NR1>|OFF|ON}

Arguments  

<NR1> = 0 disables underline font, any other value turns this feature on.

OFF disables underline font.

ON enables underline font.
Examples
PLOT:PLOT1:LABEL:FONT:UNDERline 1 set the font to underlined.
PLOT:PLOT1:LABEL:FONT:UNDERline?
:PLOT:PLOT1:LABEL:FONT:UNDERLINE 0 indicating the font is underlined.

PLOT:PLOT<x>:LABEL:NAME

This command sets or queries the specified trend's label. This command/query only applies to Time Trend plots.

Group
Plot

Syntax
PLOT:PLOT<x>:LABEL:NAME <QString>

Arguments
<QString> is the label.

Examples
PLOT:PLOT1:LABEL:NAME "Time Trend number 1" sets the plot name to Time Trend number 1.
PLOT:PLOT1:LABEL:NAME? might return :PLOT:PLOT1:LABEL:NAME "This is a label test plot" indicating the label name is This is a label test plot.

PLOT:PLOT<x>:LABEL:XPOS

This command sets or queries the x-position of the specified trend label. This command/query only applies to Time Trend plots.

Group
Plot

Syntax
PLOT:PLOT<x>:LABEL:XPOS <NR3>

Arguments
<NR3> is the y-position, in pixels relative to the left edge of the display, of the label.

Examples
PLOT:PLOT1:LABEL:XPOS 200 sets the X position to 200.
PLOT:PLOT1:LABEL:XPOS? might return :PLOT:PLOT1:LABEL:XPOS 45.0000 indicating the X position is 45.0 pixels to the right of the left edge of the display.
**PLOT:PLOT<x>:LABel:YPOS**

This command sets or queries the y-position of the specified trend label. This command/query only applies to Time Trend plots.

**Group**
Plot

**Syntax**
PLOT:PLOT<x>:LABel:YPOS <NR3>

**Arguments**
<NR3> is the x-position, in pixels relative to the baseline of the waveform, of the label.

**Examples**
PLOT:PLOT1:LABel:YPOS 100 sets the Y position to 100.
PLOT:PLOT1:LABel:YPOS? might return :PLOT:PLOT1:LABEL:YPOS 0.0E+0 indicating the Y position of the label is at the baseline of the waveform.

**PLOT:PLOT<x>:MASK? (Query Only)**

This command returns the name of the mask test associated with the specified eye diagram plot.

**Group**
Plot

**Syntax**
PLOT:PLOT<x>:MASK?

**Arguments**
PLOT<x> is the plot number.

**Returns**
<String> is a quoted string that contains the name of the eye diagram mask test associated with the specified plot. If a plot does not have an eye diagram mask test, the command returns an empty string "".

**Examples**
PLOT:PLOT1:MASK? might return "mask2", indicating that the eye diagram mask test mask2 is associated with plot 1.

**PLOT:PLOT<x>:NUMBins**

This command sets or queries the current histogram resolution.
Commands listed in alphabetical order

Group    Plot

Syntax    PLOT:PLOT<x>:NUMBins  \{TWENTyfive|FIfty|HUNDred|TWOFifty|FIVEHundred|TWOTHousand|MAXimum\}

Arguments    Arguments are the number of bins.

Examples    PLOT:PLOT4:NUMBins  TWENTYFIVE sets the number of bins to 25.

PLOT:PLOT<x>:SOUrce<x>

This command sets or queries the plot source.

Group    Plot

Syntax    PLOT:PLOT<x>:SOURCE<x>  MEAS<x>  PLOT:PLOT<x>:SOURCE<x>?

Arguments    MEAS<x> is the specified measurement source for the specified plot.

Examples    PLOT:PLOT1:SOURCE1  MEAS2  sets source 1 of plot 1 to measurement 2.
PLOT:PLOT1:SOURCE1?  might return :PLOT:PLOT1:SOURCE1  MEAS1 indicating the specified source of the specified plot is measurement 1.

PLOT:PLOT<x>:SPECtrum:BASE

This command sets or queries the spectrum base. Undefined for non-spectrum plots.

Group    Plot

Syntax    PLOT:PLOT<x>:SPECTrum:BASE  <NR1>  PLOT:PLOT<x>:SPECTrum:BASE?

Arguments    <NR1> is the spectrum base.
Examples

PLOT:PLOT3:SPECTRum:BASE -10 sets the base to -10.


PLOT:PLOT<x>:SPECTrume:DYNRange

This command sets or queries the dynamic range value.

Group Plot

Syntax PLOT:PLOT<x>:SPECTrume:DYNRange <NR3>
PLOT:PLOT<x>:SPECTrume:DYNRange?

Arguments <NR3> is the dynamic range value.

Examples PLOT:PLOT3:SPECTRum:DYNRange 150 sets the dynamic range to 150 dB.

PLOT:PLOT3:SPECTRum:DYNRange? might return :PLOT:PLOT3:SPECTRum:DYNRange 100 indicating the dynamic range is 100 dB.

PLOT:PLOT<x>:TYPe

This command sets or returns the current plot type of the specified plot.

Group Plot

Syntax PLOT:PLOT<x>:TYPe {NONE|BATHTUB|EYEDIAGRAM|HARMONICS|HISTOGRAM|INDUCTANCE|IVSINTEGRALV|MAGPROPERTY|PHASENOISE|SOA|SPECTRUM|SSCPROFILE|SWL|TIEHISTOGRAM|TIETIME|TIME|TIESPECTRUM|TIME|XY|XYZ}

Arguments <x> is the plot number. This is the equivalent of the number shown on a plot heading in the UI.

NONE does not create a plot.

BATHTUB creates a bathtub plot.

EYEDIAGRAM creates an eye diagram.
HARMONICS creates a harmonics bar graph
HISTOGRAM creates a histogram plot.
INDUCTANCE creates an inductance plot.
IVSINTEGRAL creates a I vs. ∫V plot.
PHASENOISE creates a phase noise plot.
MAGPROPERTY creates a BH curve.
SOA creates an SOA plot.
SPECTRUM creates a spectrum plot.
SSCPROFILE creates a SSC profile plot.
SWL creates a Switching Loss plot.
TIEHISTOGRAM creates a TIE histogram plot.
TIESPECTRUM creates a TIE spectrum plot.
TIETIMETREND creates a TIE time trend plot.
TIMETREND creates a time trend plot.
XY creates an XY plot.
XYZ creates an XYZ plot.

Examples

PLOT:PLOT2:TYPE BATHTUB creates a bathtub plot.
PLOT:PLOT2:TYPE? might return :PLOT:PLOT2:TYPE EYEDIAGRAM indicating the plot is an eye diagram.

POWer:ADDNew (No Query Form)

This command adds the specified power measurement number. The power measurement number is specified by <x>.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:ADDNew “POWER<x>”

Examples

POWER:ADDNEW “POWER1” adds POWER1 measurement badge and selects Switching Loss measurement by default.
**POWer:DELeTe (No Query Form)**

This command deletes the specified power measurement number. The power measurement number is specified by <x>.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:DELeTe "POWER<x>"

**Examples**
POWer:DELeTe "POWER1" deletes the POWER1 measurement badge.

**POWer:POWer<x>:AUTOSet (No Query Form)**

This command executes power autoset for the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:AUTOSet EXECute

**Arguments**
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

**Examples**
POWer:POWer1:AUTOSet EXECute executes the power autoset for the power measurement Power 1.

**POWer:POWer<x>:CLRESPONSE:AMP[1-10]Val**

This command sets or queries the generator amplitude value of the specified configuration step for the Control Loop Response power measurement.

**Conditions**
Requires option 5-PWR or 6-PWR
Commands listed in alphabetical order

Group Power

Syntax
POWeR:POWeRx:CLRESPONSE:AMP[1-10]Val <NR3>
POWeR:POWeRx:CLRESPONSE:AMP[1-10]Val?

Arguments
Power<x> sets the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
AMP[1–10] sets the configuration step number, in the range of 1 to 10. Values outside this range will report an error.
<NR3> sets the generator amplitude for the specified configuration step, in the range of –100 V to 100 V.

Examples
POWeR:POWeR1:CLRESPONSE:AMP3Val 20 sets the generator output amplitude for configuration step 3 to 20 volts, for power measurement 1.
POWeR:POWeR2:CLRESPONSE:AMP8Val? might return 60, indicating that the generator output amplitude setting of configuration step 8 is 60 volts, for power measurement 2.

POWeR:POWeRx:CLRESPONSE:AMPMode

This command sets or queries the amplitude mode for the Control Loop Response power measurement.

Conditions
Requires option 5-PWR or 6-PWR

Group Power

Syntax
POWeR:POWeRx:CLRESPONSE:AMPMode {CONSTant|PROFile}
POWeR:POWeRx:CLRESPONSE:AMPMode?

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
CONSTant sets the amplitude mode to output a constant amplitude signal from the DUT stimulus generator for all frequency bands.
PROFile enables configuring the generator to set amplitude values for each frequency band.
**Examples**

POWER:POWER1:CLRESPONSE:AMPMode CONSTANT sets the amplitude mode to Constant for power measurement 1.

POWER:POWER2:CLRESPONSE:AMPMode? might return PROFILE, indicating that the amplitude mode power measurement 2 is set to Profile.

**POWER:Power<x>:CLRESPONSE:CONSTAMPlitude**

This command sets or queries the constant amplitude voltage for the Control Loop Response power measurement.

**Conditions**
Requires option 5-PWR or 6-PWR

**Group**
Power

**Syntax**

```
POWER:POWER<x>:CLRESPONSE:CONSTAMPlitude <NR3>
POWER:POWER<x>:CLRESPONSE:CONSTAMPlitude?
```

**Arguments**

<power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR3> is the constant amplitude voltage value for the measurement, in the range of –100 V to 100 V.

**Examples**

POWER:POWER3:CLRESPONSE:CONSTAMPlitude 120 sets the constant amplitude voltage for Power measurement 3 to be 120 volts.

POWER:POWER5:CLRESPONSE:CONSTAMPlitude? might return –15, indicating that the constant amplitude voltage for power measurement 5 is –15 volts.


This command sets or queries the generator frequency value of the specified configuration step for the Control Loop Response power measurement.

**Conditions**
Requires option 5-PWR or 6-PWR

**Group**
Power

**Syntax**

```
```
Commands listed in alphabetical order

Arguments

Power<x> sets the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

FREQ[1–11] sets the configuration step number, in the range of 1 to 11. Values outside this range will report an error.

<NR3> sets the frequency of the specified configuration step number, in the range of 10 Hz to 50 MHz.

Examples

POW<er>:POW<er><x>:CLRESPONSE:FREQ1Val 200 sets the generator frequency value for frequency band 1 to 200 Hz, for power measurement 5.

POW<er>:POW<er><x>:CLRESPONSE:FREQ3Val? might return 2.000E+6, indicating that the generator frequency output for frequency band 3 is 2.0 MHz, for power measurement 2.

POW<er>:POW<er><x>:CLRESPONSE:GENerator

This command sets or queries the generator source used to send stimulus signals to the DUT, for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR

Group

Power

Syntax

POW<er>:POW<er><x>:CLRESPONSE:GENerator {INTernal}
POW<er>:POW<er><x>:CLRESPONSE:GENerator?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

INTERNAL sets the generator to the oscilloscope AFG. This is the only valid argument.

Examples

POW<er>:POW<er>3:CLRESPONSE:GENerator INTERNAL sets the generator to the oscilloscope AFG for power measurement 3.

POW<er>:POW<er>2:CLRESPONSE:GENerator? might return INTERNAL, indicating that the generator source for power measurement 2 is the oscilloscope AFG.
POWer:POWer<x>:CLRESPONSE:IMPEdance

This command sets or queries the vertical termination impedance for the Control Loop Response power measurement.

**Conditions**
Requires option 5-PWR or 6-PWR

**Group**
Power

**Syntax**
```
POWer:POWer<x>:CLRESPONSE:IMPEdance {FIFTy|ONEMEGa} 
POWer:POWer<x>:CLRESPONSE:IMPEdance?
```

**Arguments**
`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

- **FIFTy** sets the impedance to be 50 Ω.
- **ONEMEGa** sets the impedance to be 1 MΩ.

**Examples**
```
POWer:POWer3:CLRESPONSE:IMPEdance FIFTy sets the vertical termination impedance for power measurement 3 to be 50 Ω.
POWer:POWer2:CLRESPONSE:IMPEdance? might return CH6, indicating that the vertical termination impedance for power measurement 2 is 1 MΩ.
```

POWer:POWer<x>:CLRESPONSE:INPUTSource

This command sets or queries the input source for the Control Loop Response power measurement.

**Conditions**
Requires option 5-PWR or 6-PWR

**Group**
Power

**Syntax**
```
POWer:POWer<x>:CLRESPONSE:INPUTSource CH<x> 
POWer:POWer<x>:CLRESPONSE:INPUTSource?
```

**Arguments**
`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

- **CH<x>** sets the channel to use for the output signal source.
POWer:POWer<x>:CLRESPONSE:INPUTSOurce

POWer:POWer2:CLRESPONSE:INPUTSOurce CH8 sets the input for power measurement 2 to be the channel 8 waveform.

POWer:POWer2:CLRESPONSE:INPUTSOurce? might return CH1, indicating that the input source for power measurement 1 is channel 1.

POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce

This command sets or queries the output source for the Control Loop Response power measurement.

Conditions
Requires option 5-PWR or 6-PWR

Group
Power

Syntax
POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce CH<x>
POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce?

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

CH<x> sets the channel to use for the output signal source.

Examples
POWer:POWer1:CLRESPONSE:OUTPUTSOurce CH5 sets the output source for power measurement 1 to be the channel 5 waveform.

POWer:POWer3:CLRESPONSE:OUTPUTSOurce? might return CH6, indicating that the output source for power measurement 3 is channel 6.

POWer:POWer<x>:CLRESPONSE:PPD

This command sets or queries the points per decade (PPD) value for the Control Loop Response power measurement.

Conditions
Requires option 5-PWR or 6-PWR

Group
Power

Syntax
POWer:POWer<x>:CLRESPONSE:PPD <NR3>
POWer:POWer<x>:CLRESPONSE:PPD?
Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR3> is the PPD value for the measurement, in the range of 10 to 100 points.

Examples

POWer:POWer3:CLRESPONSE:PPD 30 sets the PPD for Power measurement 3 to be 30 points.

POWer:POWer5:CLRESPONSE:PPD? might return 20, indicating that the PPD value for power measurement 5 is 20 points.

POWer:POWer<x>:CLRESPONSE:STARTFREQuency

This command sets or queries the start frequency value for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR

Group

Power

Syntax

POWer:POWer<x>:CLRESPONSE:STARTFREQuency <NR3>

POWer:POWer<x>:CLRESPONSE:STARTFREQuency?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR3> is the starting frequency for the measurement, in the range of 10 Hz to 50 MHz.

Examples

POWer:POWer8:CLRESPONSE:STARTFREQuency 12 sets the starting frequency for power measurement 8 to be 12 Hz.

POWer:POWer3:CLRESPONSE:STARTFREQuency? might return 2400, indicating that the starting frequency for power measurement 3 is 2400 Hz.

POWer:POWer<x>:CLRESPONSE:STOPFREQuency

This command sets or queries the stop frequency value for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR
Commands listed in alphabetical order

Group  Power

Syntax  POWER:POWer<x>:CLRESPONSE:STOPFREQuency  <NR3>
        POWER:POWer<x>:CLRESPONSE:STOPFREQuency?

Arguments  Power<x>  is the power measurement number. This is the equivalent of the 
           number shown in the UI for a power measurement badge.
           <NR3>  is the stop frequency for the measurement, in the range of 10 Hz to 50 MHz.

Examples  POWER:POWer3:CLRESPONSE:STOPFREQuency  120 sets the stop frequency 
           for Power measurement 8 to be 120 Hz.
           POWER:POWer5:CLRESPONSE:STOPFREQuency?  might return 2000, indicating
           that the stop frequency for power measurement 5 is 2000 Hz.

POWER:POWer<x>:CYCLEAmp:INPUTSOurse

This command sets or queries the input source for cycle amplitude measurement 
of the specified power measurement number.

Conditions  Requires option PWR or PS2.

Group  Power

Syntax  POWER:POWer<x>:CYCLEAmp:INPUTSOurse  {CH<x>|MATH<x>|REF<x>}
        POWER:POWer<x>:CYCLEAmp:INPUTSOurse?

Arguments  Power<x>  is the power measurement number. This is the equivalent of the 
           number shown on a power measurement badge in the UI.
           CH<x>  = A channel specifier; <x> is 1 through 8 and is limited by the number 
           of FlexChannels in your instrument.
           MATH<x>  = A math waveform specifier; <x> is $\geq$1.
           REF<x>  = A reference waveform specifier; <x> is $\geq$1.

Examples  POWER:POWer1:CYCAmp:INPUTSOurse  CH1 sets the input source for cycle 
           amplitude measurement as CH1 for the power measurement badge Power 1.
POWer:POWer<x>:CYCLEBase:INPUTSOurce

This command sets or queries the input source for cycle base measurement of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:CYCLEBase:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLEBase:INPUTSOurce?

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier, <x> is ≥1.

REF<x> = A reference waveform specifier, <x> is ≥1.

Examples
POWer:POWer1:CYCLEBase:INPUTSOurce CH2 sets the input source for cycle base measurement as CH2 for the power measurement badge Power 1.

POWer:POWer<x>:CYCLEMAX:INPUTSOurce

This command sets or queries the input source for cycle maximum measurement in the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:CYCLEMAX:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLEMAX:INPUTSOurce?

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.
Commands listed in alphabetical order

**MATH<x>** = A math waveform specifier; <x> is ≥1.

**REF<x>** = A reference waveform specifier; <x> is ≥1.

**Examples**

POWer:POWer1:CYCMAX:INPUTSOurce CH2 sets the input source for cycle maximum measurement as CH2 for the power measurement badge Power 1.

**POWer:POWer<x>:CYCLEMin:INPUTSOurce**

This command sets or queries the input source for cycle minimum measurement in the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```plaintext
POWer:POWer<x>:CYCLEMin:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLEMin:INPUTSOurce?
```

**Arguments**

- **Power<x>** is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
- **CH<x>** = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.
- **MATH<x>** = A math waveform specifier; <x> is ≥1.
- **REF<x>** = A reference waveform specifier; <x> is ≥1.

**Examples**

POWer:POWer1:CYCMIn:INPUTSOurce CH2 sets the input source for cycle minimum measurement as CH2 for the power measurement badge Power 1.

**POWer:POWer<x>:CYCLEPKPK:INPUTSOurce**

This command sets or queries the input source for cycle peak-to-peak measurement in the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power
Syntax

`POWer:POWer<x>:CYCLEPKPK:INPUTSOurce {CH<x>|MATH<x>|REF<x>}`

`POWer:POWer<x>:CYCLEPKPK:INPUTSOurce?`

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; `<x>` is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; `<x>` is ≥1.

`REF<x>` = A reference waveform specifier; `<x>` is ≥1.

Examples

`POWer:POWer1:CYCLEPKPK:INPUTSOurce REF1` sets the input source for cycle peak-to-peak measurement as REF1 for the power measurement badge Power 1.

`POWer:POWer<x>:CYCLETop:INPUTSOurce {CH<x>|MATH<x>|REF<x>}`

`POWer:POWer<x>:CYCLETop:INPUTSOurce?`

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; `<x>` is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; `<x>` is ≥1.

`REF<x>` = A reference waveform specifier; `<x>` is ≥1.

Examples

`POWer:POWer1:CYCLETop:INPUTSOurce REF3` sets the input source for cycle top measurement as REF3 for the power measurement badge Power 1.
POWer:POWer<x>:DIDT:INPUTSOurce

This command sets or queries the input source for di/dt measurement in the specified power measurement number.

**Conditions** Requires option PWR or PS2.

**Group** Power

**Syntax**
- `POWER:POWer<x>:DIDT:INPUTSOurce {CH<x>|MATH<x>|REF<x>}`
- `POWER:POWer<x>:DIDT:INPUTSOurce?`

**Arguments**
- `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
- `CH<x>` = A channel specifier; `<x>` is 1 through 8 and is limited by the number of FlexChannels in your instrument.
- `MATH<x>` = A math waveform specifier; `<x>` is ≥1.
- `REF<x>` = A reference waveform specifier; `<x>` is ≥1.

**Examples**
- `POWER:POWer1:DIDT:INPUTSOurce CH6` sets the input source for di/dt measurement as channel 2.

POWer:POWer<x>:DIDT:SOURCEEDGEType

This command sets or queries the edge type for di/dt measurement of the specified power measurement number. `<x>` specifies the number of the power measurement badge.

**Conditions** Requires option PWR or PS2.

**Group** Power

**Syntax**
- `POWER:POWer<x>:DIDT:SOURCEEDGEType {RISE|FALL}`
- `POWER:POWer<x>:DIDT:SOURCEEDGEType?`

**Examples**
- `POWER:POWer1:DIDT:SOURCEEDGEType RISE` sets the edge type as rise for the di/dt measurement.
POWer:POWer<x>:DVDT:INPUTSOurce

This command sets or queries the input source for dv/dt measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:DVDT:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:DVDT:INPUTSOurce?

**Arguments**
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier, <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier, <x> is ≥1.

REF<x> = A reference waveform specifier, <x> is ≥1.

**Examples**
POWer:POWer1:DVDT:INPUTSOurce CH1 sets the input source for dv/dt measurement as channel 1.

POWer:POWer<x>:DVDT:SOURCEEDGEType

This command sets or queries the edge type for dv/dt measurement in the specified power measurement number. The power measurement number is specified by <x>.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:DVDT:SOURCEEDGEType {RISE|FALL}
POWer:POWer<x>:DVDT:SOURCEEDGEType?

**Examples**
POWer:POWer1:DVDT:SOURCEEDGEType RISE sets the edge type as rise for the dv/dt measurement.
POWer:POWer<x>:EFFICIENCY:INPUTType

This command sets or queries the input type (AC or DC) for power Efficiency measurement of the specified power measurement number.

Conditions   Requires option PWR or PS2.

Group        Power

Syntax

POWer:POWer<x>:EFFICIENCY:INPUTType {AC|DC}
POWer:POWer<x>:EFFICIENCY:INPUTType?

Related Commands
POWer:POWer<x> EFFICIENCY:OUTPUT1Type

Arguments

Power<x> is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge of the UI.

AC sets the input voltage type to AC.
DC sets the input voltage type to DC.

Examples

POWer:POWer1:EFFICIENCY:INPUTType DC sets the input voltage type to DC for power Efficiency measurement 1
POWer:POWer2:EFFICIENCY:INPUTType? might return AC, indicating that the input voltage type for power Efficiency measurement 2 is AC.

POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce

This command sets or queries the output 1 current source for the power Efficiency measurement of the specified power measurement number.

Conditions   Requires option PWR or PS2.

Group        Power

Syntax

POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce?
Related Commands

POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce
POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce

Arguments

Power<x> is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:EFFICIENCY:IOUT1SOUrce CH4 sets the source for the output 1 current efficiency measurement of power measurement 1 to Channel 4.

POWer:POWer1:EFFICIENCY:IOUT1SOUrce? might return CH6, indicating that the source for the output 1 current efficiency measurement of power measurement 1 is Channel 6.

POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce

This command sets or queries the output 2 current source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce {CH<x>|MATH<x>|REF<x>}  
POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce?

Related Commands

POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce
POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce

Arguments

Power<x> is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.
Commands listed in alphabetical order

REF\textsubscript{<x>} = A reference waveform specifier; \( <x> \) is \( \geq 1 \).

Examples

POWer:POWer3:EFFICIENCY:IOUT2SOURCE \textsubscript{REF2} sets the source for the output 2 current efficiency measurement of power measurement 3 to Reference waveform 2.

POWer:POWer1:EFFICIENCY:IOUT2SOURCE? might return MATH1, indicating that the source for the output 2 current efficiency measurement of power measurement 1 is Math waveform 1.

POWer:POWer\textsubscript{<x>}:EFFICIENCY:IOUT3SOURCE

This command sets or queries the output 3 current source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer\textsubscript{<x>}:EFFICIENCY:IOUT3SOURCE \{CH\textsubscript{<x>} | MATH\textsubscript{<x>} | REF\textsubscript{<x>} \}

POWer:POWer\textsubscript{<x>}:EFFICIENCY:IOUT3SOURCE?

Related Commands

POWer:POWer\textsubscript{<x>}:EFFICIENCY:IOUT1SOURCE

POWer:POWer\textsubscript{<x>}:EFFICIENCY:IOUT2SOURCE

Arguments

Power\textsubscript{<x>} is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge in the UI.

CH\textsubscript{<x>} = A channel specifier; \( <x> \) is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH\textsubscript{<x>} = A math waveform specifier; \( <x> \) is \( \geq 1 \).

REF\textsubscript{<x>} = A reference waveform specifier; \( <x> \) is \( \geq 1 \).

Examples

POWer:POWer1:EFFICIENCY:IOUT3SOURCE \textsubscript{REF2} sets the source for the output 3 current efficiency measurement of power measurement 1 to Reference waveform 2.

POWer:POWer4:EFFICIENCY:IOUT3SOURCE? might return CH8, indicating that the source for the output 3 current efficiency measurement of power measurement 4 is Channel 8.
POWer:POWer<x>:EFFICIENCY:ISOUrce

This command sets or queries the current source for the power Efficiency measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:EFFICIENCY:ISOUrce
POWer:POWer<x>:EFFICIENCY:ISOUrce?

**Related Commands**
POWer:POWer<x>:EFFICIENCY:VSOUrce

**Arguments**
Power<x> is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier, <x> is ≥1.

REF<x> = A reference waveform specifier, <x> is ≥1.

**Examples**
POWer:POWer1:EFFICIENCY:ISOUrce CH2 sets the input current source for power Efficiency measurement 1 to Channel 2.

POWer:POWer6:EFFICIENCY:ISOUrce? might return CH2, indicating that the input current source for power Efficiency measurement 6 is Channel 2.

POWer:POWer<x>:EFFICIENCY:NUMOFOutputs

This command sets or queries the number of outputs for the power Efficiency measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:EFFICIENCY:NUMOFOutputs {ONE|TWO|THREE}
POWer:POWer<x>:EFFICIENCY:NUMOFOutputs?

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer

2-605
Related Commands

POWer:POWer<x>:EFFICIENCY:OUCt1SOUrce
POWer:POWer<x>:EFFICIENCY:OUCt2SOUrce
POWer:POWer<x>:EFFICIENCY:OUCt3SOUrce
POWer:POWer<x>:EFFICIENCY:VOUT1SOUrce
POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce
POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce

Arguments

Power<x> is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

ONE, TWO, THREE sets the number of outputs to test in the power efficiency measurement.

Examples

POWer:POWer4:EFFICIENCY:NUMOFOutputs TWO sets the number of outputs to measure in power measurement 4 to two.

POWer:POWer3:EFFICIENCY:NUMOFOutputs? might return ONE, indicating that the number of outputs being measured in power measurement 3 is one.

POWer:POWer<x>:EFFICIENCY:OUTPUT1Type

This command sets or queries the Output1 type (AC or DC) for the power efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:EFFICIENCY:OUTPUT1Type {AC|DC}
POWer:POWer<x>:EFFICIENCY:OUTPUT1Type?

Related Commands

POWer:POWer<x>:EFFICIENCY:INPUTType

Arguments

Power<x> is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge of the UI.

AC sets the output1 voltage type to AC.

DC sets the output1 voltage type to DC.
Examples

POWer:POWER1:EFFICIENCY:OUTPUT1Type DC sets the output type to DC for power Efficiency measurement 1.

POWer:POWER3:EFFICIENCY:OUTPUT1Type? might return AC, indicating that the power output type for power Efficiency measurement 3 is AC.

POWer:POWER<x>:EFFICIENCY:OUTPUT2Type

This command sets or queries the Output2 type (AC or DC) for the power Efficiency measurement of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWER<x>:EFFICIENCY:OUTPUT2Type {AC|DC}
POWer:POWER<x>:EFFICIENCY:OUTPUT2Type?

Related Commands
POWer:POWER<x>:EFFICIENCY:INPUTType

Arguments
Power<x> is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge of the UI.
AC sets the Output2 voltage type to AC.
DC sets the Output2 voltage type to DC.

Examples

POWer:POWER1:EFFICIENCY:OUTPUT2Type DC sets the Output2 type to DC for power Efficiency measurement 1.

POWer:POWER3:EFFICIENCY:OUTPUT2Type? might return AC, indicating that the power Output2 type for power Efficiency measurement 3 is AC.

POWer:POWER<x>:EFFICIENCY:OUTPUT3Type

This command sets or queries the Output3 type (AC or DC) for the power Efficiency measurement of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power
Syntax

POWer:POWer<x>:EFFICIENCY:OUTPUT3Type {AC|DC}
POWer:POWer<x>:EFFICIENCY:OUTPUT3Type?

Related Commands

POWer:POWer<x>:EFFICIENCY:INPUTType

Arguments

Power<x> is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge of the UI.

AC sets the Output3 voltage type to AC.
DC sets the Output3 voltage type to DC.

Examples

POWer:POWer1:EFFICIENCY:OUTPUT3Type DC sets the Output3 type to DC for power Efficiency measurement 1.
POWer:POWer3:EFFICIENCY:OUTPUT3Type? might return AC, indicating that the power Output3 type for power Efficiency measurement 3 is AC.

POWer:POWer<x>:EFFICIENCY:OUTPUTType

This command sets or queries the output type (AC or DC) for power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:EFFICIENCY:OUTPUTType {AC|DC}
POWer:POWer<x>:EFFICIENCY:OUTPUTType?

Related Commands

POWer:POWer<x>:EFFICIENCY:INPUTType

Arguments

Power<x> is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge of the UI.

AC sets the output voltage type to AC.
DC sets the output voltage type to DC.

Examples

POWer:POWer1:EFFICIENCY:OUTPUTType DC sets the output type to DC for power Efficiency measurement 1.
POWer:POWer3:EFFICIENCY:OUTPUTType? might return AC, indicating that the power output type for power Efficiency measurement 3 is AC.

POWer:POWer<x>:EFFICIENCY:VOUT1SOURce

This command sets or queries the output 1 voltage source for the power Efficiency measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:EFFICIENCY:VOUT1SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:EFFICIENCY:VOUT1SOURce?

**Related Commands**
POWer:POWer<x>:EFFICIENCY:VOUT2SOURce
POWer:POWer<x>:EFFICIENCY:VOUT3SOURce

**Arguments**
Power<x> is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> = A channel specifier, <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier, <x> is ≥1.

REF<x> = A reference waveform specifier, <x> is ≥1.

**Examples**
POWer:POWer4:EFFICIENCY:VOUT1SOURce CH4 sets the source for the output 1 voltage efficiency measurement of power measurement 4 to Channel 4.

POWer:POWer1:EFFICIENCY:VOUT1SOURce? might return CH3, indicating that the source for the output 1 voltage efficiency measurement of power measurement 1 is Channel 3.

POWer:POWer<x>:EFFICIENCY:VOUT2SOURce

This command sets or queries the output 2 voltage source for the power Efficiency measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.
Commands listed in alphabetical order

**Group**
Power

**Syntax**
```
POWer:POWER<x>:EFFICIENCY:VOUT2Source {CH<x>|MATH<x>|REF<x>}
POWer:POWER<x>:EFFICIENCY:VOUT2Source?
```

**Related Commands**
```
POWer:POWER<x>:EFFICIENCY:VOUT1Source
POWer:POWER<x>:EFFICIENCY:VOUT3Source
```

**Arguments**
- `Power<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.
- `CH<x>` = A channel specifier; `<x>` is 1 through 8 and is limited by the number of FlexChannels in your instrument.
- `MATH<x>` = A math waveform specifier; `<x>` is ≥1.
- `REF<x>` = A reference waveform specifier; `<x>` is ≥1.

**Examples**
```
POWer:POWER4:EFFICIENCY:VOUT2Source CH5 sets the source for the output 2 voltage efficiency measurement of power measurement 4 to Channel 5.

POWer:POWER1:EFFICIENCY:VOUT2Source? might return CH3, indicating that the source for the output 2 voltage efficiency measurement of power measurement 1 is Channel 3.
```

**POWer:POWER<x>:EFFICIENCY:VOUT3Source**

This command sets or queries the output 3 voltage source for the power Efficiency measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
```
POWer:POWER<x>:EFFICIENCY:VOUT3Source {CH<x>|MATH<x>|REF<x>}
POWer:POWER<x>:EFFICIENCY:VOUT3Source?
```

**Related Commands**
```
POWer:POWER<x>:EFFICIENCY:VOUT1Source
POWer:POWER<x>:EFFICIENCY:VOUT2Source
```
Arguments

- **Power\(<x>\)** is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge in the UI.
- **CH\(<x>\)** = A channel specifier; \(<x>\) is 1 through 8 and is limited by the number of FlexChannels in your instrument.
- **MATH\(<x>\)** = A math waveform specifier; \(<x>\) is \(\geq 1\).
- **REF\(<x>\)** = A reference waveform specifier; \(<x>\) is \(\geq 1\).

Examples

- `POWer:POWer4:EFFICIENCY:VOUT3SOUrce REF7` sets the source for the output 3 voltage efficiency measurement of power measurement 4 to Reference waveform 7.
- `POWer:POWer1:EFFICIENCY:VOUT3SOUrce?` might return `CH5`, indicating that the source for the output 3 voltage efficiency measurement of power measurement 1 is Channel 5.

**POWer:POWer\(<x>:EFFICIENCY:VSOUrce**

This command sets or queries the voltage source for the power Efficiency measurement of the specified power measurement number.

**Syntax**

- `POWer:POWer\(<x>:EFFICIENCY:VSOUrce`  
- `POWer:POWer\(<x>:EFFICIENCY:VSOUrce?`  

**Related Commands**

- `POWer:POWer\(<x>:EFFICIENCY:ISOUrce`
POWER:POWER1:EFFICIENCY:VSource? might return CH3, indicating that the input voltage source for power Efficiency measurement 1 is Channel 3.

**POWER:POWER<x>:FREQUENCY:EDGE**

This command sets or queries the edge type for frequency measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
```
POWER:POWER<x>:FREQUENCY:EDGE {RISE|FALL}
POWER:POWER<x>:FREQUENCY:EDGE?
```

**Related Commands**
`POWER:POWER<x>:FREQUENCY:INPUTSource`

**Arguments**
`Power<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

**Examples**
```
POWER:POWER1:FREQUENCY:EDGE FALL sets the edge type as fall for the frequency measurement.
```

**POWER:POWER<x>:FREQUENCY:INPUTSource**

This command sets or queries the input source for frequency measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
```
POWER:POWER<x>:FREQUENCY:INPUTSource {CH<x>|MATH<x>|REF<x>}
POWER:POWER<x>:FREQUENCY:INPUTSource?
```

**Related Commands**
`POWER:POWER<x>:FREQUENCY:EDGE`
Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier, <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier, <x> is ≥1.

REF<x> = A reference waveform specifier, <x> is ≥1.

Examples

POWer:POWer1:FREQUENCY:INPUTSource REF1 sets the input source for frequency measurement as REF1.

POWer:POWer<x>:GATing

This command sets or queries the gating type for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:GATing {NONE|CURSOR|SCREEN|LOGIC}

POWer:POWer<x>:GATing?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

NONE makes measurement across the entire waveform record.

CURSOR makes measurements on that portion of the waveform between the cursors. Selecting Cursors opens cursors on the measurement source. Set the cursors so that the waveform area of interest is in between the cursors.

SCREEN takes measurements on that portion of the waveform shown in the display. When Zoom is on, the display is the zoom window.

LOGIC takes measurements only when the logical state of a specified waveform is true.

Examples

POWer:POWer1:GATing CURSOR sets the gating type for the power measurement badge 1 as Cursor.

POWer:POWer2:GATing? might return SCREEN indicating the configured gating type for the power measurement badge 2.
POWer:POWer<x>:GATing:GLOBal

This command sets or queries the gating settings for the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:GATing:GLOBal {ON|OFF|1|0}
POWer:POWer<x>:GATing:GLOBal?

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

1 | ON selects the gating settings as Global.
0 | OFF selects the gating settings as Local.

Examples
POWer:POWer1:GATing:GLOBal 1 selects the gating settings as Global for the power measurement badge 1.

POWer:POWer<x>:HARMONICS:CLASs

This command sets or queries the class type for the harmonics measurement in the specified power measurement number. The power measurement number is specified by <x>.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:HARMONICS:CLASS {CLASSA | CLASSB | CLASSC | CLASSD}
POWer:POWer<x>:HARMONICS:CLASS?

Related Commands
POWer:POWer<x>:HARMONICS:STANDard

Examples
POWer:POWer1:HARMONICS:CLASS CLASSB sets the class type for harmonics measurement as CLASSB.
POWer:POWer<x>:HARMONICS:CMEThod

This command sets or queries the fundamental current method for the harmonics measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:HARMONICS:CMEThod {RATed | MEASured}
POWer:POWer<x>:HARMONICS:CMEThod?

**Related Commands**
POWer:POWer<x>:HARMONICS:STANDard

**Arguments**
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

- RATed: select to use the standard input current values in the measurement.
- MEASured: select to use the measured input current values in the measurement.

**Examples**
POWer:POWer1:HARMONICS:CMEThod RATed sets the current method for harmonics measurement as rated.

POWer:POWer<x>:HARMONICS:FUNDCURRent

This command sets or queries the fundamental current value for the harmonics measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:HARMONICS:FUNDCURRent <NR1>
POWer:POWer<x>:HARMONICS:FUNDCURRent?

**Related Commands**
POWer:POWer<x>:HARMONICS:CLASs
Arguments

Power\textsubscript{x} is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.<x1>

\textsubscript{NR1} ranges from 0 to 16.

Examples

\texttt{POWer:POWer\textsubscript{x}:HARMONICS:FUNDCURRENT 1.5} sets the fundamental current for the harmonics measurement as 1.5.

\textbf{POWer:POWer\textsubscript{x}:HARMONICS:HORDer}

This command sets or queries the order value for the harmonics measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

\texttt{POWer:POWer\textsubscript{x}:HARMONICS:HORDer \textsubscript{NR1}}

\texttt{POWer:POWer\textsubscript{x}:HARMONICS:HORDer?}

Arguments

Power\textsubscript{x} is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.<x1>

\textsubscript{NR1} ranges from 40 to 100.

Examples

\texttt{POWer:POWer1:HARMONICS:HORDer 70} sets the order value for the harmonics measurement as 70.

\textbf{POWer:POWer\textsubscript{x}:HARMONICS:HSOURCe}

This command sets or queries the source type for the harmonics measurement of the specified power measurement number. The power measurement number is specified by \textsubscript{x}.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

\texttt{POWer:POWer\textsubscript{x}:HARMONICS:HSOURCe \{CURRENT \mid VOLTage\}}

\texttt{POWer:POWer\textsubscript{x}:HARMONICS:HSOURCe?}
Related Commands

POWer:POWer<x>:HARMONICS:STANDard

Examples

POWer:POWer1:HARMONICS:HSOURce CURRENT sets the source as current for the harmonics measurement.

POWer:POWer<x>:HARMONICS:IPOWer

This command sets or queries the input power value for the harmonics measurement of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax

POWer:POWer<x>:HARMONICS:IPOWer <NR1>

POWer:POWer<x>:HARMONICS:IPOWer?

Related Commands
POWer:POWer<x>:HARMONICS CLASs

Arguments

<NR1> ranges from 0 to 600.

Examples

POWer:POWer<x>:HARMONICS:IPOWer 150 sets the input power for the harmonics measurement as 150.

POWer:POWer<x>:HARMONICS:ISOURce

This command sets or queries the current source for SOA measurement of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax

POWer:POWer<x>:HARMONICS:ISOURce {CH<x>|MATH<x>|REF<x>}

POWer:POWer<x>:HARMONICS:ISOURce?
Arguments

- **Power<x>** is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

- **CH<x>** = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

- **MATH<x>** = A math waveform specifier; <x> is \( \geq 1 \).

- **REF<x>** = A reference waveform specifier; <x> is \( \geq 1 \).

Examples

```
POWer:POWer1:HARMONICS:ISOURce CH2 sets the current source for harmonics measurement as channel 2.
```

**POWer:POWer<x>:HARMONICS:LINEFREQUEncy**

This command sets or queries the value for the line frequency for the Harmonics measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:LINEFREQUEncy {Auto|FIFTyhz|SIXTyhz |THREESIXTyhz|FOURHUNDReDhz|SIXFIFTyhz|EIGHTHUNDReDhz|CUSTom
```

```
POWer:POWer<x>:HARMONICS:LINEFREQUEncy?
```

Related Commands

- **POWer:POWer<x>:HARMONICS:LINEFREQUency**

Arguments

- **Power<x>** is the Power measurement identifier number.

  - **Auto** automatically detects and sets the line frequency value.

  - **FIFTyhz** sets the line frequency value to 50 Hz.

  - **SIXTyhz** sets the line frequency value to 60 Hz.

  - **THREESIXTyhz** sets the line frequency value to 360 Hz.

  - **FOURHUNDReDhz** sets the line frequency value to 400 Hz.

  - **SIXFIFTyhz** sets the line frequency value to 650 Hz.

  - **EIGHTHUNDReDhz** sets the line frequency value to 800 Hz.

  - **CUSTom** sets the line frequency value to Custom. The default value for custom is 100 Hz. Use the **POWer:POWer<x>:HARMONICS:LINEFREQUEncy** command to set a custom line frequency value.
Examples

POWer:POWer3:HARMONICS:LINEFREQUENCY FIFTyHz sets the line frequency value for the Harmonics power measurement 3 to 50 Hz.

POWer:POWer1:HARMONICS:LINEFREQUENCY? might return POWer:POWer1:HARMONICS:LINEFREQUENCY CUSTOM, indicating that the line frequency setting for Harmonics power measurement 1 is a custom value.

POWer:POWer<x>:HARMONICS:ODDEVen

This command sets or queries the harmonics value analysis format of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:HARMONICS:ODDEVen {ALL | ODD | EVEN}
POWer:POWer<x>:HARMONICS:ODDEVen?

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

ALL to display all harmonics values
ODD to display only the odd values of harmonics
EVEN to display only the even values of harmonics

Examples
POWer:POWer3:HARMONICS:ODDEVen ALL sets the harmonics value analysis format to display all result values for harmonics measurement 3.

POWer:POWer<x>:HARMONICS:PFACtor

This command sets or queries the value of power factor for the harmonics measurement of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power
Commands listed in alphabetical order

Syntax

POWer:POWer<x>:HARMONICS:PFACtor <NR1>
POWer:POWer<x>:HARMONICS:PFACtor?

Related Commands

POWer:POWer<x>:HARMONICS:CLASs

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR1> ranges from 0 to 1.

Examples

POWer:POWer2:HARMONICS:PFACtor 0.7 sets the power factor as 0.7 for harmonics measurement 2.

**POWer:POWer<x>:HARMONICS:POWERRating**

This command sets or queries the power level for the harmonics measurement of the specified power measurement number. The power measurement number is specified by <x>.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:HARMONICS:POWERRating {HIGH | LOW}
POWer:POWer<x>:HARMONICS:POWERRating?

Related Commands

POWer:POWer<x>:HARMONICS:STANDard

Examples

POWer:POWer1:HARMONICS:POWERRating HIGH sets the power level as high for harmonics measurement 1.

**POWer:POWer<x>:HARMONICS:RCURRent**

This command sets or queries the rated current for the harmonics measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.
Group: Power

Syntax:

```
POWer:POWer<x>:HARMONICS:RCURRent <NR1>
POWer:POWer<x>:HARMONICS:RCURRent?
```

Related Commands:

```
POWer:POWer<x>:HARMONICS:CMEThod
```

Arguments:

- `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
- `<NR1>` ranges from 0 to 100

Examples:

```
POWer:POWer2:HARMONICS:RCURRent 1.5 sets the rated current as 1.5 for harmonics measurement 2.
```

### POWer:POWer<x>:HARMONICS:STANDard

This command sets or queries the test mode for harmonics measurement of the specified power measurement number.

Conditions:

- Requires option PWR or PS2.

Arguments:

- `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
- `NONE` = No standard
- `IEC` = IEC 61000-3-2 standard
- `MIL` = MIL-STD-1399 standard
- `AM14` = AM14 standard

Examples:

```
POWer:POWer3:HARMONICS:STANDard AM14 sets the test mode for harmonics measurement as AM14 for harmonics measurement 3.
```
Commands listed in alphabetical order

**POWer:POWer<x>:HARMONICS:STARTFREQUEncy**

This command sets or queries the value for the start frequency for the Harmonics measurement in the range of 1 Hz to 1 GHz.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:HARMONICS:STARTFREQUEncy <NR2>
POWer:POWer<x>:HARMONICS:STARTFREQUEncy?
```

**Arguments**

- `POWer<x>` is the Power measurement.
- `<NR2>` sets the starting frequency, in hertz.

**Examples**

```
POWer:POWer1:HARMONICS:STARTFREQUEncy 1000
```
sets the Harmonics measurement start frequency to 1 kHz for power measurement 1.

```
POWer:POWer4:HARMONICS:STARTFREQUEncy? might return
POWer:POWer4:HARMONICS:STARTFREQUEncy 60 indicating that the start frequency for Harmonics power measurement 4 is 60 Hz.
```

**POWer:POWer<x>:HARMONICS:UNITs**

This command sets or queries the harmonics results units of the specified power measurement number. The power measurement number is specified by `<x>`.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:HARMONICS:UNITs {LOG | LINear}
POWer:POWer<x>:HARMONICS:UNITs?
```

**Examples**

```
POWer:POWer1:HARMONICS:UNITs LOG
```
sets the harmonics results units as logarithmic for harmonics measurement 1.
POWer:POWer<x>:HARMONICS:VSOURce

This command sets or queries the voltage source for SOA measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:HARMONICS:VSOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:HARMONICS:VSOURce?

**Arguments**
- Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
- CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.
- MATH<x> = A math waveform specifier; <x> is ≥1.
- REF<x> = A reference waveform specifier; <x> is ≥1.

**Examples**
POWer:POWer4:HARMONICS:VSOURce CH1 sets the voltage source for harmonics measurement 4 as channel 1.

POWer:POWer<x>:INDUCTANCE:EDGESource

This command sets or queries the edge source for the power inductance measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:INDUCTANCE:EDGESource {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:INDUCTANCE:EDGESource?

**Related Commands**
POWer:POWer<x>:INDUCTANCE:ISOURce
POWer:POWer<x>:INDUCTANCE:VSOURce
Arguments

Power<\textgreater x\textless> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

\texttt{CH<x>} = A channel specifier; \texttt{<x>} is 1 through 8 and is limited by the number of FlexChannels in your instrument.

\texttt{MATH<x>} = A math waveform specifier; \texttt{<x>} is \geq 1.

\texttt{REF<x>} = A reference waveform specifier; \texttt{<x>} is \geq 1.

Examples

\texttt{POWer:POWer1:HARMONICS:INDUCTANCE:EDGESource CH2} sets the signal edge source for inductance measurement 1 as Channel 2.

\texttt{POWer:POWer3:HARMONICS:INDUCTANCE:EDGESource?} might return CH4, indicating that the signal edge source for inductance measurement 3 is Channel 4.

\texttt{POWer:POWer<x>:INDUCTANCE:ISOUrce} sets or queries the current signal source for the inductance measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

\texttt{POWer:POWer<x>:INDUCTANCE:ISOUrce}

\texttt{POWer:POWer<x>:INDUCTANCE:ISOUrce?}

Related Commands

\texttt{POWer:POWer<x>:INDUCTANCE:EDGESource}

\texttt{POWer:POWer<x>:INDUCTANCE:VSOURce}

Arguments

Power<\textgreater x\textless> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

\texttt{CH<x>} = A channel specifier; \texttt{<x>} is 1 through 8 and is limited by the number of FlexChannels in your instrument.

\texttt{MATH<x>} = A math waveform specifier; \texttt{<x>} is \geq 1.

\texttt{REF<x>} = A reference waveform specifier; \texttt{<x>} is \geq 1.

Examples

\texttt{POWer:POWer7:HARMONICS:INDUCTANCE:ISOUrce CH3} sets the current signal source for inductance measurement 7 as Channel 3.
POWER:POWER3:HARMONICS:INDUCTANCE:ISource? might return CH4, indicating that the current signal source for inductance measurement 3 is Channel 4.

**POWER:POWER<x>:INDUCTANCE:VSOURCE**

This command sets or queries the voltage source for inductance measurement of the specified power measurement number.

**Conditions** Requires option PWR or PS2.

**Group** Power

**Syntax**

```
POWER:POWER<x>:INDUCTANCE:VSOURCE
POWER:POWER<x>:INDUCTANCE:VSOURCE?
```

**Related Commands**

- POWER:POWER<x>:INDUCTANCE:EDGESource
- POWER:POWER<x>:INDUCTANCE:ISOURce

**Arguments**

- **Power<x>** is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
- **CH<x>** = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.
- **MATH<x>** = A math waveform specifier; <x> is ≥1.
- **REF<x>** = A reference waveform specifier; <x> is ≥1.

**Examples**

- POWER:POWER7:HARMONICS:INDUCTANCE:VSOURCE CH3 sets the voltage signal source for inductance measurement 7 as Channel 3.
- POWER:POWER1:HARMONICS:INDUCTANCE:VSOURCE? might return MATH2, indicating that the voltage signal source for inductance measurement 1 is Math waveform 2.

**POWER:POWER<x>:INPUTCAP:ISOURce**

This command sets or queries the inrush current input source of the specified Input Capacitance measurement.

**Conditions** Requires option PWR or PS2.
Group  Power

Syntax  POWER:POWer<x>:INPUTCAP:ISOURce {CH<x>|REF<x>|MATH<x>}
        POWER:POWer<x>:INPUTCAP:ISOURce?

Arguments  Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> is the channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> is the Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> is the Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples  POWER:POWER1:INRUSHcurrent:ISOURce REF3 sets the current source of Input Capacitance 1 to Reference waveform 3.


POWER:POWER<x>:INPUTCAP:PEAKCURRent

This command sets or queries the peak current value of the specified Input Capacitance measurement.

Conditions  Requires option PWR or PS2.

Group  Power

Syntax  POWER:POWER<x>:INPUTCAP:PEAKCURRent <NR3>
        POWER:POWER<x>:INPUTCAP:PEAKCURRent?

Arguments  Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the peak current value, in amps, in the range -100 A to 100 A.
Examples

POWer:POWer4:INPUTCAP:PEAKCURRent 12.5 sets the peak current value of Input Capacitance measurement 4 to 12.5 amps.

POWer:POWer2:INPUTCAP:PEAKCURRent? might return POWer:POWer2:INPUTCAP:PEAKCURRent −85, indicating that the peak current setting of Input Capacitance measurement 2 is −85 amps.

POWer:POWer<x>:INPUTCAP:PEAKVOLTage

This command sets or queries the peak voltage value of the specified Input Capacitance measurement.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:INPUTCAP:PEAKVOLTage <NR3>
POWer:POWer<x>:INPUTCAP:PEAKVOLTage?

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the peak voltage value in the range -100 V to 100 V.

Examples

POWer:POWer4:INPUTCAP:PEAKVOLTage 122.5 sets the peak voltage value of Input Capacitance measurement 4 to 122.5 volts.

POWer:POWer2:INPUTCAP:PEAKVOLTage? might return POWer:POWer2:INPUTCAP:PEAKVOLTage 955, indicating that the peak voltage setting of Input Capacitance measurement 2 is 955 volts.

POWer:POWer<x>:INPUTCAP:VSOURce

This command sets or queries the input voltage source of the specified Input Capacitance measurement.

Conditions
Requires option PWR or PS2.

Group
Power
Commands listed in alphabetical order

Syntax

POWe:r:POWe<r>:INRUSHcurrent:INPUTSOurce

POWe:r:POWe<r>:INRUSHcurrent:INPUTSOurce?

Arguments

Power<r> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<r> is the channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<r> is the Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<r> is the Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWe:r:POWe1:INRUSHcurrent:INPUTSOurce CH2 sets the voltage source of the input capacitance measurement 1 to Reference Channel 2.

POWe:r:POWe1:INRUSHcurrent:INPUTSOurce? might return

POWe:r:POWe2:INRUSHcurrent:INPUTSOurce MATH6, indicating that the voltage source of Input Capacitance measurement 2 is Math waveform 6.

POWe:r:POWe<r>:INRUSHcurrent:INPUTSOurce

This command sets or returns the input source of the specified Inrush Current measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWe:r:POWe<r>:INRUSHcurrent:INPUTSOurce

POWe:r:POWe<r>:INRUSHcurrent:INPUTSOurce?

Arguments

Power<r> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<r> is the channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<r> is the Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<r> is the Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.
Examples

```
POWer:POWer3:INRUSHcurrent:INPUTSOurce MATH1 sets the input source of Inrush Current measurement 3 to Math waveform 1.

POWer:POWer2:INRUSHcurrent:INPUTSOurce? might return
POWer:POWer2:INRUSHcurrent:INPUTSOurce CH7, indicating that the input source of Inrush Current measurement 2 is Channel 7.
```

### POWer:POWer<x>:INRUSHCurrent:PEAKCURRENT

This command sets or returns the peak current value of the specified Inrush Current measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:INRUSHcurrent:PEAKCURRENT <NR3>
POWer:POWer<x>:INRUSHcurrent:PEAKCURRENT?
```

**Arguments**

- `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

- `<NR3>` is a floating point number that represents the peak current value, in amps, in the range -100 A to 100 A.

**Examples**

```
POWer:POWer4:INRUSHcurrent:PEAKCURRENT 12.5 sets the peak current value of Inrush Current measurement 4 to 12.5 amps.

POWer:POWer2:INRUSHcurrent:PEAKCURRENT? might return
POWer:POWer2:INRUSHcurrent:PEAKCURRENT 75, indicating that the peak current setting of Inrush Current measurement 2 is 75 amps.
```

### POWer:POWer<x>:IVSINTEGRALV:ISOURce

This command sets or queries the current source for I vs Integral V measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power
Commands listed in alphabetical order

Syntax

POWer:POWer<x>:IVSINTEGRALV:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:IVSINTEGRALV:ISOURce?

Related Commands

POWer:POWer<x>:IVSINTEGRALV:VSOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:IVSINTEGRALV:ISOURce CH7 sets power measurement 1 to use Channel 7 as the current signal source for the measurement.

POWer:POWer2:IVSINTEGRALV:ISOURce? might return CH1, indicating that channel 1 is the current signal source for power measurement number 2.

POWer:POWer<x>:IVSINTEGRALV:VSOURce

This command sets or queries the voltage source for I vs Integral V measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:IVSINTEGRALV:VSOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:IVSINTEGRALV:VSOURce?

Related Commands

POWer:POWer<x>:IVSINTEGRALV:ISOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.
Examples

POWer:POWer2:IVSINTEGRALV:VSOURce CH1 sets I vs Integral V power measurement 2 to use Channel 1 as the voltage signal source for the measurement.

POWer:POWer4:IVSINTEGRALV:VSOURce? might return CH1, indicating that channel 1 is the voltage source for I vs Integral V power measurement number 4.

POWer:POWer<x>:LABel

This command sets or queries the label for the specified power measurement. As the label can contain non 7-bit ASCII text, it is stored in Percent Encoding format. The power measurement badge is specified by x.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:LABel <QString>
POWer:POWer<x>:LABel?

Examples
POWer:POWer1:LABel "NewMeasurement" sets the custom measurement name for the measurement in Power 1 badge as New Measurement.

POWer:POWer<x>:LINERIPPLE:INPUTSOurce

This command sets or queries the input source for line ripple measurement of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:LINERIPPLE:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:LINERIPPLE:INPUTSOurce?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.
Commands listed in alphabetical order

REF<x> = A reference waveform specifier, <x> is ≥1.

Examples
- POWER:POWER1:LINERIPPLE:INPUTSOURcE CH1 sets channel 1 as the input source for line ripple measurement of power measurement badge Power 1.

**POWER:POWER<x>:LINERIPPLE:LFREQuency**

This command sets or queries the frequency present for line ripple measurement of the specified power measurement number. The power measurement number is specified by <x>.

Conditions
- Requires option PWR or PS2.

Group
- Power

Syntax
- POWER:POWER<x>:LINERIPPLE:LFREQuency {FIFTy | SIXty | FOURHundred}
- POWER:POWER<x>:LINERIPPLE:LFREQuency?

Examples
- POWER:POWER2:LINERIPPLE:LFREQuency FIFTY sets the frequency present for line ripple measurement as 50 Hz for power measurement 2.

**POWER:POWER<x>:MAGNETICLOSS:ISOURce**

This command sets or queries the current source for the magnetic loss measurement of the specified power measurement number.

Conditions
- Requires option PWR or PS2.

Group
- Power

Syntax
- POWER:POWER<x>:MAGNETICLOSS:ISOURce {CH<x> | MATH<x> | REF<x>}
- POWER:POWER<x>:MAGNETICLOSS:ISOURce?

Related Commands
- POWER:POWER<x>:MAGNETICLOSS:VSOURce

Arguments
- Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.
CH<x> sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

**Examples**

POWER:POWER1:MAGNETICLOSS:ISOURce CH7 sets power measurement 1 to use Channel 7 as the current signal source for the magnetic loss measurement.

POWER:POWER2:MAGNETICLOSS:ISOURce? might return CH1, indicating that channel 1 is the current signal source for the magnetic loss measurement number 2.

**POWER:POWER<x>:MAGNETICLOSS:VSOURce**

This command sets or queries the voltage source for magnetic measurement of the specified power measurement number.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

`POWER:POWER<x>:MAGNETICLOSS:VSOURce {CH<x>|MATH<x>|REF<x>}`

`POWER:POWER<x>:MAGNETICLOSS:VSOURce?`

**Related Commands**

POWER:POWER<x>:MAGNETICLOSS:ISOURce

**Arguments**

Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

**Examples**

POWER:POWER2:MAGNETICLOSS:VSOURce CH4 sets power measurement 2 to use Channel 4 as the voltage signal source for the magnetic loss measurement.

POWER:POWER1:MAGNETICLOSS:VSOURce? might return CH1, indicating that channel 1 is the voltage source for the magnetic loss measurement number 1.
POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection

This command sets or queries the coil cross section area for magnetic measurement of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection <NR2>
POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection?

Related Commands
POWer:POWer<x>:MAGPROPERTY:UNITs

Arguments
Power<x> is the magnetic property power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

<NR2> is the cross section area in square meters, in the range of 1 nanometer$^2$ to 1 M$^2$.

Examples
POWer:POWer2:MAGPROPERTY:AREAofcrosssection .005 sets the primary winding cross section area to 5 for power measurement 2.

POWer:POWer4:MAGPROPERTY:AREAofcrosssection? might return 10.54E-3, indicating that the coil cross section area for power measurement 4 is 0.01054.

NOTE. The coil Cross Section Area unit depends on the setting of the Units value (see POWer:POWer<x>:MAGPROPERTY:UNITs). The command returns the same numeric value regardless of the units setting.

POWer:POWer<x>:MAGPROPERTY:EDGESOURce

This command sets or queries the edge source type for the magnetic property measurement of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power
**POWer:POWer<x>:MAGPROPERTY:EDGESOURce**

Syntax

```
POWer:POWer<x>:MAGPROPERTY:EDGESOURce {Current | VOLTAGE}
POWer:POWer<x>:MAGPROPERTY:EDGESOURce?
```

Arguments

- **Power<x>** is the magnetic property power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
- **Current** sets the measurement to use the primary voltage source as the signal edge for the magnetic property measurement.
- **VOLTAGE** sets the measurement to use the primary current source as the signal edge for the magnetic property measurement.

Examples

```
POWer:POWer1:MAGPROPERTY:EDGESource VOLTAGE sets the signal edge source for power measurement 1 to use the primary voltage source.
POWer:POWer3:MAGPROPERTY:EDGESource? might return CURRENT, indicating that the signal edge source for power measurement 3 is the primary current source.
```

**POWer:POWer<x>:MAGPROPERTY:ISOURce**

Syntax

```
POWer:POWer<x>:MAGPROPERTY:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:ISOURce?
```

Arguments

- **Power<x>** is the magnetic property power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.
- **CH<x>** sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.
- **MATH<x>** = A math waveform specifier, <x> is ≥1.
- **REF<x>** = A reference waveform specifier, <x> is ≥1.
Examples

POWer:POWer1:MAGPROPERTY:ISOURce CH4 sets power measurement 1 to use channel 4 as the primary winding current source for the magnetic power measurement.

POWer:POWer2:MAGPROPERTY:ISOURce? might return CH1, indicating that channel 1 is the primary winding current source for the magnetic power measurement number 2.

**POWer:POWer<x>:MAGPROPERTY:LENgth**

This command sets or queries the conductor length of the primary winding for magnetic measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

POWer:POWer<x>:MAGPROPERTY:LENgth <NR2>
POWer:POWer<x>:MAGPROPERTY:LENgth?

**Related Commands**

POWer:POWer<x>:MAGPROPERTY:UNITs

**Arguments**
Power<x> is the magnetic property power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

<NR2> is the magnetic length, in the range of 1.00E-09 through 1,000,000.

**Examples**

POWer:POWer2:MAGPROPERTY:LENgth 2.5 sets the primary winding length to 2.5 for power measurement 2.

POWer:POWer5:MAGPROPERTY:LENgth? might return 10.54E-3, indicating that the primary winding length for power measurement 5 is 0.01054.

**NOTE.** The Magnetic Length unit depends on the setting of the Units value (see **POWer:POWer<x>:MAGPROPERTY:UNITs**). The command returns the same numeric value regardless of the units setting.

**POWer:POWer<x>:MAGPROPERTY:PRIMARYTURNs**

This command sets or queries the number of primary turns for magnetic measurement of the specified power measurement number.
Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:MAGPROPERTY:PRIMARYTURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:PRIMARYTURNs?

Related Commands
POWer:POWer<x>:MAGPROPERTY:LENgth
POWer:POWer<x>:MAGPROPERTY:AREAofcrossection

Arguments
Power<x> is the magnetic property power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

<NR1> is the integer number of turns in the primary winding, in the range of 1 to 1 M.

Examples
POWer:POWer2:MAGPROPERTY:PRIMARYTURNs 150 sets the primary winding turn count to 150 for power measurement 2.

POWer:POWer5:MAGPROPERTY:PRIMARYTURNs? might return 50, indicating that the primary winding has 50 turns for power measurement 5.

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce

This command sets or queries the current source channel for secondary winding 1 for magnetic measurement of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:MAGPROPERTY:SEC1SOURce {CH<x> | MATH<x> | REF<x>} 
POWer:POWer<x>:MAGPROPERTY:SEC1SOURce?

Related Commands
POWer:POWer<x>:MAGPROPERTY:SEC2SOURce
POWer:POWer<x>:MAGPROPERTY:SEC3SOURce
POWer:POWer<x>:MAGPROPERTY:SEC4SOURce
POWer:POWer<x>:MAGPROPERTY:SEC5SOURce
Commands listed in alphabetical order

**POWer:POWer<x>:MAGPROPERTY:SEC6SOURce**

**Arguments**
- **Power<x>** is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.
- **CH<x>** sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.
- **MATH<x>** = A math waveform specifier; <x> is ≥1.
- **REF<x>** = A reference waveform specifier; <x> is ≥1.

**Examples**
- `POWer:POWer3:MAGPROPERTY:SEC1SOURce CH7` sets power measurement number 3 to use channel 7 as the current source for measuring secondary winding 1.
- `POWer:POWer1:MAGPROPERTY:SEC1SOURce?` might return CH3, indicating that channel 3 is the source for measuring secondary winding 1 of magnetic power measurement number 1.

**POWer:POWer<x>:MAGPROPERTY:SEC1TURNs**

This command sets or queries the number of turns of secondary winding 1 for magnetic measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:MAGPROPERTY:SEC1TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC1TURNs?
```

**Related Commands**
- `POWer:POWer<x>:MAGPROPERTY:SEC2TURNs`
- `POWer:POWer<x>:MAGPROPERTY:SEC3TURNs`
- `POWer:POWer<x>:MAGPROPERTY:SEC4TURNs`
- `POWer:POWer<x>:MAGPROPERTY:SEC5TURNs`
- `POWer:POWer<x>:MAGPROPERTY:SEC6TURNs`

**Arguments**
- **Power<x>** is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.
<NR1> is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

**Examples**

POWER:POWER3:MAGPROPERTY:SEC1TURNS 500 sets the number of turns on secondary winding 1 to 500 for magnetic power measurement number 3.

POWER:POWER3:MAGPROPERTY:SEC1TURNS? might return 2500, indicating that there are 2,500 turns on secondary winding 4 for magnetic power measurement number 3.

**POWer:POWer<x>:MAGPROPERTY:SEC2SOURce**

This command sets or queries the current source for secondary winding2 for magnetic measurement of the specified power measurement number.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce {CH<x>|MATH<x>|REF<x>}

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce?

**Related Commands**

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce

POWer:POWer<x>:MAGPROPERTY:SEC4SOURce

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce

**Arguments**

Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

**Examples**

POWER:POWer1:MAGPROPERTY:SEC2SOURce CH2 sets power measurement number 1 to use channel 2 as the current source for measuring secondary winding 2.
POWer:POWer3:MAGPROPERTY:SEC2SOURce? might return MATH2, indicating that math waveform 2 is the source for measuring secondary winding 2 of magnetic power measurement number 3.

POWer:POWer<x>:MAGPROPERTY:SEC2TURNs

This command sets or queries the number of turns of secondary winding 2 for magnetic measurement of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:MAGPROPERTY:SEC2TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC2TURNs?

Related Commands
POWer:POWer<x>:MAGPROPERTY:SEC1TURNs
POWer:POWer<x>:MAGPROPERTY:SEC3TURNs
POWer:POWer<x>:MAGPROPERTY:SEC4TURNs
POWer:POWer<x>:MAGPROPERTY:SEC5TURNs
POWer:POWer<x>:MAGPROPERTY:SEC6TURNs

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

<NR1> is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples
POWer:POWer1:MAGPROPERTY:SEC2TURNs 50 sets the number of turns on secondary winding 2 to 50 for magnetic power measurement number 3.

POWer:POWer7:MAGPROPERTY:SEC2TURNs? might return 250, indicating that there are 250 turns on secondary winding 2 for magnetic power measurement number 7.

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce

This command sets or queries the current source channel for secondary winding 3 for magnetic measurement of the specified power measurement number.
Commands listed in alphabetical order

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:MAGPROPERTY:SEC3SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:SEC3SOURce?
```

**Related Commands**

- POWer:POWer<x>:MAGPROPERTY:SEC1SOURce
- POWer:POWer<x>:MAGPROPERTY:SEC2SOURce
- POWer:POWer<x>:MAGPROPERTY:SEC4SOURce
- POWer:POWer<x>:MAGPROPERTY:SEC5SOURce
- POWer:POWer<x>:MAGPROPERTY:SEC6SOURce

**Arguments**

- **Power<x>** is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.
- **CH<x>** sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.
- **MATH<x>** = A math waveform specifier; <x> is ≥1.
- **REF<x>** = A reference waveform specifier; <x> is ≥1.

**Examples**

```
POWer:POWer1:MAGPROPERTY:SEC3SOURce CH1 sets power measurement number 1 to use channel 1 as the current source for measuring secondary winding 3.
POWer:POWer1:MAGPROPERTY:SEC3SOURce? might return CH1, indicating that channel 1 is the source for measuring secondary winding 3 of magnetic power measurement number 1.
```

**POWer:POWer<x>:MAGPROPERTY:SEC3TURNs**

This command sets or queries the number of turns of secondary winding 3 for magnetic measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power
Commands listed in alphabetical order

### Syntax

**POWER:**POWER<\(x\)>:MAGPROPERTY:SEC3TURNs \(<\text{NR1}\>\)
**POWER:**POWER<\(x\)>:MAGPROPERTY:SEC3TURNs?

### Related Commands

- **POWER:**POWER<\(x\)>:MAGPROPERTY:SEC1TURNs
- **POWER:**POWER<\(x\)>:MAGPROPERTY:SEC2TURNs
- **POWER:**POWER<\(x\)>:MAGPROPERTY:SEC4TURNs
- **POWER:**POWER<\(x\)>:MAGPROPERTY:SEC5TURNs
- **POWER:**POWER<\(x\)>:MAGPROPERTY:SEC6TURNs

### Arguments

- **Power<\(x\)>** is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.
- **<\(\text{NR1}\)>** is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

### Examples

- **POWER:**POWER1:MAGPROPERTY:SEC3TURNs 15 sets the number of turns on secondary winding 3 to 15 for magnetic power measurement number 1.
- **POWER:**POWER3:MAGPROPERTY:SEC3TURNs? might return 2500, indicating that there are 2,500 turns on secondary winding 3 for magnetic power measurement number 3.

### **POWER:**POWER<\(x\)>:MAGPROPERTY:SEC4SOURce

This command sets or queries the current source for secondary winding 4 for magnetic measurement of the specified power measurement number.

### Conditions

Requires option PWR or PS2.

### Group

Power

### Syntax

**POWER:**POWER<\(x\)>:MAGPROPERTY:SEC4SOURce \{CH<\(x\)> | MATH<\(x\)> | REF<\(x\)>\}

**POWER:**POWER<\(x\)>:MAGPROPERTY:SEC4SOURce?

### Related Commands

- **POWER:**POWER<\(x\)>:MAGPROPERTY:SEC1SOURce
- **POWER:**POWER<\(x\)>:MAGPROPERTY:SEC2SOURce
- **POWER:**POWER<\(x\)>:MAGPROPERTY:SEC3SOURce
- **POWER:**POWER<\(x\)>:MAGPROPERTY:SEC5SOURce

2-642 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
POWer:POWer<x>:MAGPROPERTY:SEC6SOURce

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples
POWer:POWer3:MAGPROPERTY:SEC4SOURce CH7 sets power measurement number 3 to use channel 7 as the current source for measuring secondary winding 4.

POWer:POWer12:MAGPROPERTY:SEC4SOURce? might return CH1, indicating that Channel 1 is the source for measuring secondary winding 4 of magnetic power measurement number 12.

POWer:POWer<x>:MAGPROPERTY:SEC4TURNs

This command sets or queries the number of turns of secondary winding 4 for magnetic measurement of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:MAGPROPERTY:SEC4TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC4TURNs?

Related Commands
POWer:POWer<x>:MAGPROPERTY:SEC1TURNs
POWer:POWer<x>:MAGPROPERTY:SEC2TURNs
POWer:POWer<x>:MAGPROPERTY:SEC3TURNs
POWer:POWer<x>:MAGPROPERTY:SEC5TURNs
POWer:POWer<x>:MAGPROPERTY:SEC6TURNs

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.
<NR1> is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples

POWer:POWer4:MAGPROPERTY:SEC4TURNS 500 sets the number of turns on secondary winding 4 to 500 for magnetic power measurement number 4.

POWer:POWer1:MAGPROPERTY:SEC4TURNS? might return 90, indicating that there are 90 turns on secondary winding 4 for magnetic power measurement number 3.

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce

This command sets or queries the current source for secondary winding 5 for magnetic measurement of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce {CH<x>|MATH<x>|REF<x>}

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce?

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce
POWer:POWer<x>:MAGPROPERTY:SEC2SOURce
POWer:POWer<x>:MAGPROPERTY:SEC3SOURce
POWer:POWer<x>:MAGPROPERTY:SEC4SOURce
POWer:POWer<x>:MAGPROPERTY:SEC6SOURce

Arguments

Power <x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer3:MAGPROPERTY:SEC5SOURce MATH1 sets power measurement number 3 to use math waveform 1 as the current source for measuring secondary winding 5.
POWer:POWer4:MAGPROPERTY:SEC5SOURce? might return CH2, indicating that channel 2 is the source for measuring secondary winding 5 of magnetic power measurement number 4.

POWer:POWer<x>:MAGPROPERTY:SEC5TURNs

This command sets or queries the number of turns of secondary winding 5 for magnetic measurement of the specified power measurement badge.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:MAGPROPERTY:SEC5TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC5TURNs?

**Related Commands**
POWer:POWer<x>:MAGPROPERTY:SEC1TURNs
POWer:POWer<x>:MAGPROPERTY:SEC2TURNs
POWer:POWer<x>:MAGPROPERTY:SEC3TURNs
POWer:POWer<x>:MAGPROPERTY:SEC4TURNs
POWer:POWer<x>:MAGPROPERTY:SEC6TURNs

**Arguments**
Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

<NR1> is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

**Examples**
POWer:POWer3:MAGPROPERTY:SEC5TURNs 250 sets the number of turns on secondary winding 5 to 250 for magnetic power measurement number 3.

POWer:POWer1:MAGPROPERTY:SEC5TURNs? might return 150, indicating that there are 150 turns on secondary winding 5 for magnetic power measurement number 1.

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce

This command sets or queries the current source for secondary winding 6 for magnetic measurement of the specified power measurement number.
Commands listed in alphabetical order

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
```
POWer:POWer<x>:MAGPROPERTY:SEC6SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:SEC6SOURce?
```

**Related Commands**
- `POWer:POWer<x>:MAGPROPERTY:SEC1SOURce`
- `POWer:POWer<x>:MAGPROPERTY:SEC2SOURce`
- `POWer:POWer<x>:MAGPROPERTY:SEC3SOURce`
- `POWer:POWer<x>:MAGPROPERTY:SEC4SOURce`
- `POWer:POWer<x>:MAGPROPERTY:SEC5SOURce`

**Arguments**
- `Power<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.
- `CH<x>` sets the channel specifier; `<x>` is 1 through 8 and is limited by the number of FlexChannels in your instrument.
- `MATH<x>` = A math waveform specifier; `<x>` is ≥1.
- `REF<x>` = A reference waveform specifier; `<x>` is ≥1.

**Examples**
- `POWer:POWer3:MAGPROPERTY:SEC6SOURce CH4` sets power measurement number 3 to use channel 4 as the current source for measuring secondary winding 6.
- `POWer:POWer2:MAGPROPERTY:SEC6SOURce?` might return CH4, indicating that channel 4 is the source for measuring secondary winding 3 of magnetic power measurement number 2.

**POWer:POWer<x>:MAGPROPERTY:SEC6TURNs**

This command sets or queries the number of turns of secondary winding 6 for magnetic measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power
Syntax

POWer:POWer<x>:MAGPROPERTY:SEC6TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC6TURNs?

Related Commands
POWer:POWer<x>:MAGPROPERTY:SEC1TURNs
POWer:POWer<x>:MAGPROPERTY:SEC2TURNs
POWer:POWer<x>:MAGPROPERTY:SEC3TURNs
POWer:POWer<x>:MAGPROPERTY:SEC4TURNs
POWer:POWer<x>:MAGPROPERTY:SEC5TURNs

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

<NR1> is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples

POWer:POWer8:MAGPROPERTY:SEC6TURNs 22 sets the number of turns on secondary winding 6 to 22 for magnetic power measurement number 8.

POWer:POWer9:MAGPROPERTY:SEC6TURNs? might return 7000, indicating that there are 7,000 turns on secondary winding 6 for magnetic power measurement number 9.

POWer:POWer<x>:MAGPROPERTY:SECWINDings

This command sets or queries the number of secondary windings for the magnetic property measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:MAGPROPERTY:SECWINDings {None|ONE|TWO|THREE|FOUR|FIVE|SIX}
POWer:POWer<x>:MAGPROPERTY:SECWINDings?

Related Commands
POWer:POWer<x>:MAGPROPERTY:SEC1SOURce
POWer:POWer<x>:MAGPROPERTY:SEC1TURNs
POWer:POWer<x>:MAGPROPERTY:SEC2SOURce
POWer:POWer<x>:MAGPROPERTY:SEC2TURNs
POWer:POWer<x>:MAGPROPERTY:SEC3SOURce
POWer:POWer<x>:MAGPROPERTY:SEC3TURNs
POWer:POWer<x>:MAGPROPERTY:SEC4SOURce
POWer:POWer<x>:MAGPROPERTY:SEC4TURNs
POWer:POWer<x>:MAGPROPERTY:SEC5SOURce
POWer:POWer<x>:MAGPROPERTY:SEC5TURNs
POWer:POWer<x>:MAGPROPERTY:SEC6SOURce
POWer:POWer<x>:MAGPROPERTY:SEC6TURNs

Arguments
Power<x> is the magnetic property power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

None, ONE, TWO, THREE, FOUR, FIVE, SIX sets the number of secondary windings to the specified value.

Examples
POWer:POWer2:MAGPROPERTY:SECWINDings 4 sets power measurement 2 to use 4 secondary transformer windings.
POWer:POWer11:MAGPROPERTY:SECWINDings? might return FOUR, indicating that there are four secondary transformer windings for power measurement 11.

POWer:POWer<x>:MAGPROPERTY:UNITs

This command sets or queries the units for magnetic measurements of the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:MAGPROPERTY:UNITs {SI|CGS}
POWer:POWer<x>:MAGPROPERTY:UNITs?

Related Commands
POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection
POWer:POWer<x>:MAGPROPERTY:LENght

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.
**SI** sets the measurement to International System of Units.
**CGS** sets the measurement to Gaussian units.

**Examples**

```
POWer:POWer5:MAGPROPERTY:UNITS SI
```
sets magnetic property measurement 5 to use SI units.

```
POWer:POWer1:MAGPROPERTY:UNITS?
```
might return **CGS**, indicating that the measurement unit for power measurement 1 is set to CGS.

**POWer:POWer<x>:MAGPROPERTY:VSOURce**

This command sets or queries the primary winding voltage source for the magnetic measurement of the specified power measurement number.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

```
POWer:POWer<x>:MAGPROPERTY:VSOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:VSOURce?
```

**Related Commands**

**POWer:POWer<x>:MAGPROPERTY:ISOURce**

**Arguments**

**POWer<x>** is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

**CH<x>** sets the channel specifier; **<x>** is 1 through 8 and is limited by the number of FlexChannels in your instrument.

**MATH<x>** = A math waveform specifier, **<x>** is ≥1.

**REF<x>** = A reference waveform specifier, **<x>** is ≥1.

**Examples**

```
POWer:POWer1:MAGPROPERTY:VSOURce CH4
```
sets power measurement 1 to use channel 4 as the primary winding voltage source for the magnetic power measurement.

```
POWer:POWer2:MAGPROPERTY:VSOURce?
```
might return CH1, indicating that channel 1 is the primary winding voltage source for the magnetic power measurement number 2.
POWer:POWer<x>:NDUTYCYCLE:EDGEType

This command sets or queries the clock edge type for negative duty cycle measurement in the specified power measurement number. The power measurement number is specified by <x>.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:NDUTYCYCLE:EDGEType {RISE | FALL | BOTH}
POWer:POWer<x>:NDUTYCYCLE:EDGEType?

**Examples**
POWer:POWer1:NDUTYCYCLE:EDGEType RISE sets the clock edge type as rise for the negative duty cycle measurement.

POWer:POWer<x>:NDUTYCYCLE:INPUTSOurce

This command sets or queries the input source for negative duty cycle measurement in the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:NDUTYCYCLE:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:NDUTYCYCLE:INPUTSOurce?

**Arguments**
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

**Examples**
POWer:POWer1:NDUTYCYCLE:INPUTSOurce CH3 sets the input source for negative duty cycle measurement as CH3 for the power measurement badge Power 1.
POWer:POWer<x>:NPULSEWIDTH:INPUTSOurce

This command sets or queries the input source for negative pulse width measurement in the specified power measurement number.

Conditions  Requires option PWR or PS2.

Group  Power

Syntax  

\[
\text{POWer:POWer<x>:NPULSEWIDTH:INPUTSOurce} \\
\{\text{CH<x>|MATH<x>|REF<x>}}
\]

\[
\text{POWer:POWer<x>:NPULSEWIDTH:INPUTSOurce?}
\]

Arguments  

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples  

POWer:POWer1:NPULSEWIDTH:INPUTSOurce CH1 sets the input source for negative pulse width measurement as channel 1.

POWer:POWer<x>:PDUTYCYCLE:EDGEType

This command sets or queries the clock edge type for positive duty cycle measurement in the specified power measurement number. The power measurement number is specified by <x>.

Conditions  Requires option PWR or PS2.

Group  Power

Syntax  

\[
\text{POWer:POWer<x>:PDUTYCYCLE:EDGEType} \\
\{\text{CH<x>|MATH<x>|REF<x>}}
\]

\[
\text{POWer:POWer<x>:PDUTYCYCLE:EDGEType?}
\]

Examples  

POWer:POWer1:PDUTYCYCLE:EDGEType BOTH sets the clock edge type as both (rise and fall) for the positive duty cycle measurement.
**POWER:POWER<x>:PDUTYCYCLE:INPUTSOURCE**

This command sets or queries the input source for positive duty cycle measurement in the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWER:POWER<x>:PDUTYCYCLE:INPUTSOURCE {CH<x>|MATH<x>|REF<x>}
POWER:POWER<x>:PDUTYCYCLE:INPUTSOURCE?
```

**Arguments**

- **Power<x>** is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
- **CH<x>** = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.
- **MATH<x>** = A math waveform specifier; <x> is ≥1.
- **REF<x>** = A reference waveform specifier; <x> is ≥1.

**Examples**

```
POWER:POWER1:PDUTYCYCLE:INPUTSOURCE CH4 sets the input source for positive duty cycle measurement as channel 4.
```

**POWER:POWER<x>:PERIOD:EDGE**

This command sets or queries the edge type for period measurement in the specified power measurement number. The power measurement number is specified by <x>.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWER:POWER<x>:PERIOD:EDGE {RISE | FALL}
POWER:POWER<x>:PERIOD:EDGE?
```

**Examples**

```
POWER:POWER1:PERIOD:EDGE RISE sets the edge type as rise for the period measurement.
```
POWer:POWer<x>:PERIOD:INPUTSOurce

This command sets or queries the input source for period measurement in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

POWer:POWer<x>:PERIOD:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:PERIOD:INPUTSOurce?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:PERIOD:INPUTSOurce CH6 sets the input source for period measurement as channel 6.

POWer:POWer<x>:POWERQUALITY:CCYCles

This command sets or queries the calculate cycles over full cycles settings for the specified power quality measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

POWer:POWer<x>:POWERQUALITY:CCYCles {ON | OFF 1 | 0}
POWer:POWer<x>:POWERQUALITY:CCYCles?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

1 | ON selects the calculate cycles over full cycles.
0  |  OFF unselects the calculate cycles over full cycles.

**Examples**

```
POWer:POWer1:POWERQUALITY:CCYCles 1 selects the calculate cycles over full cycles for the power measurement badge 1.
```

**POWer:POWer<x>:POWERQUALITY:FREference**

This command sets or queries the frequency reference type for power quality measurement in the specified power measurement number. The power measurement number is specified by `<x>`.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

```
POWer:POWer<x>:POWERQUALITY:FREference {VOLTage | CURRENT}
POWer:POWer<x>:POWERQUALITY:FREference?
```

**Examples**

```
POWer:POWer1:POWERQUALITY:FREference CURRENT sets the frequency reference type for power quality measurement as current.
```

**POWer:POWer<x>:POWERQUALITY:ISOURce**

This command sets or queries the current source for power quality measurement in the specified power measurement number.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

```
POWer:POWer<x>:POWERQUALITY:ISOURce {CH<x> | MATH<x> | REF<x>}
POWer:POWer<x>:POWERQUALITY:ISOURce?
```

**Arguments**

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`CH<x>` = A channel specifier; `<x>` is 1 through 8 and is limited by the number of FlexChannels in your instrument.
Commands listed in alphabetical order

\[ \text{MATH}\langle x \rangle = \text{A math waveform specifier; } \langle x \rangle \geq 1. \]
\[ \text{REF}\langle x \rangle = \text{A reference waveform specifier; } \langle x \rangle \geq 1. \]

**Examples**

\[ \text{POWer:POWer1:POWERQUALITY:VSOURce CH2} \text{ sets the current source for power quality measurement as channel 2.} \]

**POWer:POWer\langle x \rangle:POWERQUALITY:VSOURce**

This command sets or queries the voltage source for power quality measurement in the specified power measurement number.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

\[ \text{POWer:POWer}\langle x \rangle:POWERQUALITY:VSOURce \{ \text{CH}\langle x \rangle | \text{MATH}\langle x \rangle | \text{REF}\langle x \rangle \} \]
\[ \text{POWer:POWer}\langle x \rangle:POWERQUALITY:VSOURce? \]

**Arguments**

\[ \text{Power}\langle x \rangle \text{ is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.} \]
\[ \text{CH}\langle x \rangle = \text{A channel specifier; } \langle x \rangle \text{ is 1 through 8 and is limited by the number of FlexChannels in your instrument.} \]
\[ \text{MATH}\langle x \rangle = \text{A math waveform specifier; } \langle x \rangle \geq 1. \]
\[ \text{REF}\langle x \rangle = \text{A reference waveform specifier; } \langle x \rangle \geq 1. \]

**Examples**

\[ \text{POWer:POWer1:POWERQUALITY:VSOURce CH1} \text{ sets the voltage source for power quality measurement as channel 1.} \]

**POWer:POWer\langle x \rangle:PPULSEWIDTH:INPUTSOurce**

This command sets or queries the input source for positive pulse width measurement in the specified power measurement number.

**Conditions**

Requires option PWR or PS2.

**Group**

Power
**Syntax**

`POWer:POWer<x>:PPULSEWIDTH:INPUTSource {CH<x>|MATH<x>|REF<x>}`

`POWer:POWer<x>:PPULSEWIDTH:INPUTSource?`

**Arguments**

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥ 1.

`REF<x>` = A reference waveform specifier; <x> is ≥ 1.

**Examples**

`POWer:POWer1:PPULSEWIDTH:INPUTSource CH5` sets channel 5 as the input source for the positive pulse width measurement 1.

**POWer:POWer<x>:PRESET (No Query Form)**

This command runs a power preset action for the specified power measurement number.

**Conditions**

Requires option 5-PWR, 6-PWR, or PS2.

**Group**

Power

**Syntax**

`POWer:POWer<x>:PRESET {EXECute}`

**Arguments**

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`EXECute` runs the power preset action.

**Examples**

`POWer:POWer5:PRESET EXECute` runs the power preset action for power measurement 5.

**POWer:POWer<x>:PSRR:AMP[1-10]Val**

This command sets or queries the generator amplitude value of the specified configuration step for the Power Supply Rejection Ratio (PSRR) power measurement.
Commands listed in alphabetical order

**POWer:POWer<x>:PSRR:AMP[1-10]Val <NR3>**

**POWer:POWer<x>:PSRR:AMP[1-10]Val?**

**POWer:POWer<x>:PSRR:AMPMode**

### Conditions
Requires option 5-PWR or 6-PWR

### Group
Power

### Syntax
POWer:POWer<x>:PSRR:AMP[1-10]Val <NR3>
POWer:POWer<x>:PSRR:AMP[1-10]Val?

**Arguments**

Power<x> sets the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

AMP[1–10] sets the configuration step number, in the range of 1 to 10. Values outside this range will report an error.

<NR3> sets the generator amplitude for the specified configuration step, in the range of –100 V to 100 V.

**Examples**

POWer:POWer1:PSRR:AMP3Val 20 sets the generator output amplitude for configuration step 3 to 20 volts, for power measurement 1.

POWer:POWer2:PSRR:AMP8Val? might return 60, indicating that the generator output amplitude setting of configuration step 8 is 60 volts, for power measurement 2.

**POWer:POWer<x>:PSRR:AMPMode**

This command sets or queries the amplitude mode for the Power Supply Rejection Ratio (PSRR) power measurement.

### Conditions
Requires option 5-PWR or 6-PWR

### Group
Power

### Syntax
POWer:POWer<x>:PSRR:AMPMode {CONSTant|PROFile}
POWer:POWer<x>:PSRR:AMPMode?

**Arguments**

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

CONSTant sets the amplitude mode to output a constant amplitude signal from the DUT stimulus generator for all frequency bands.

PROFile enables configuring the generator to set amplitude values for each frequency band.
POWer:POWer1:PSRR:AMPMode CONSTant sets the amplitude mode to Constant for power measurement 1.

POWer:POWer2:PSRR:AMPMode? might return PROFILE, indicating that the amplitude mode power measurement 2 is set to Profile.

**POWer:POWer<x>:PSRR:CONSTAMPlitude**

This command sets or queries the constant amplitude voltage for the Power Supply Rejection Ratio (PSRR) power measurement.

**Conditions**
Requires option 5-PWR or 6-PWR

**Group**
Power

**Syntax**
POWer:POWer<x>:PSRR:CONSTAMPlitude <NR3>
POWer:POWer<x>:PSRR:CONSTAMPlitude?

**Arguments**
*x* is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR3> is the constant amplitude voltage value for the measurement, in the range of -100 V to 100 V.

**Examples**
POWer:POWer3:PSRR:CONSTAMPlitude 120 sets the constant amplitude voltage for Power measurement 3 to be 120 volts.

POWer:POWer5:PSRR:CONSTAMPlitude? might return -15, indicating that the constant amplitude voltage for power measurement 5 is -15 volts.

**POWer:POWer<x>:PSRR:FREQ[1-11]Val**

This command sets or queries the generator frequency value of the specified configuration step for the Power Supply Rejection Ratio (PSRR) power measurement.

**Conditions**
Requires option 5-PWR or 6-PWR

**Arguments**
*x* is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

**Examples**
POWer:POWer3:PSRR:FREQ[1-11]Val 120 sets the constant amplitude voltage for Power measurement 3 to be 120 volts.

Syntax

POWer:POWer<x>:PSRR:FREQ[1-11]Val?

Arguments

Power<x> sets the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

FREQ[1–11] sets the configuration step number, in the range of 1 to 11. Values outside this range will report an error.

<NR3> sets the frequency of the specified configuration step number, in the range of 10 Hz to 50 MHz.

Examples

POWer:POWer5:PSRR:FREQ1Val 200 sets the generator frequency value for frequency band 1 to 200 Hz, for power measurement 5.

POWer:POWer2:PSRR:FREQ3Val? might return 2.000E+6, indicating that the generator frequency output for frequency band 3 is 2.0 MHz, for power measurement 2.

POWer:POWer<x>:PSRR:GENerator

This command sets or queries the generator source used to send stimulus signals to the DUT, for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR or 6-PWR

Group

Power

Syntax

POWer:POWer<x>:PSRR:GENerator {INTernal}
POWer:POWer<x>:PSRR:GENerator?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

{INTernal} sets the generator to the oscilloscope AFG signal source. This is the only valid argument.

Examples

POWer:POWer3:PSRR:GENerator INTernal sets the generator to the oscilloscope AFG for power measurement 3.

POWer:POWer2:PSRR:GENerator? might return INTERNAL, indicating that the generator source for power measurement 2 is the oscilloscope AFG.
POWer:POWer<x>:PSRR:IMPEDance

This command sets or queries the vertical termination impedance for the Power Supply Rejection Ratio (PSRR) power measurement.

**Conditions**
Requires option 5-PWR or 6-PWR

**Group**
Power

**Syntax**
POWer:POWer<x>:PSRR:IMPEDance {FIFTy|ONEMEGa}
POWer:POWer<x>:PSRR:IMPEDance?

**Arguments**
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
- FIFTy sets the impedance to be $50 \ \Omega$
- ONEMEGa sets the impedance to be $1 \ \text{M}\Omega$

**Examples**
POWer:POWer3:PSRR:IMPEDance FIFTy sets the vertical termination impedance for power measurement 3 to be $50 \ \Omega$.
POWer:POWer2:PSRR:IMPEDance? might return CH6, indicating that the vertical termination impedance for power measurement 2 is $1 \ \text{M}\Omega$.

POWer:POWer<x>:PSRR:INPUTSOurce

This command sets or queries the input source for the Power Supply Rejection Ratio (PSRR) power measurement.

**Conditions**
Requires option 5-PWR or 6-PWR

**Group**
Power

**Syntax**
POWer:POWer<x>:PSRR:INPUTSOurce CH<x>
POWer:POWer<x>:PSRR:INPUTSOurce?

**Arguments**
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
- CH<x> sets the channel to use for the input source.
**POWer:POWer<x>:PSRR:INPUTSOurce**

This command sets or queries the input source for power measurement X to be the channel X waveform.

**Examples**

- `POWer:POWer2:PSRR:INPUTSOurce CH3` sets the input for power measurement 2 to be the channel 3 waveform.
- `POWer:POWer2:PSRR:INPUTSOurce?` might return CH1, indicating that the input source for power measurement 1 is channel 1.

**POWer:POWer<x>:PSRR:OUTPUTSOurce**

This command sets or queries the output source for the Power Supply Rejection Ratio (PSRR) power measurement.

**Conditions**

Requires option 5-PWR or 6-PWR

**Group**

Power

**Syntax**

- `POWer:POWer<x>:PSRR:OUTPUTSOurce CH<x>`
- `POWer:POWer<x>:PSRR:OUTPUTSOurce?`

**Arguments**

- `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
- `CH<x>` sets the channel to use for the output source.

**Examples**

- `POWer:POWer1:PSRR:OUTPUTSOurce CH3` sets the output source for power measurement 1 to be the channel 3 waveform.
- `POWer:POWer3:PSRR:OUTPUTSOurce?` might return CH6, indicating that the output source for power measurement 3 is channel 6.

**POWer:POWer<x>:PSRR:PPD**

This command sets or queries the points per decade (PPD) value for the Power Supply Rejection Ratio (PSRR) power measurement.

**Conditions**

Requires option 5-PWR or 6-PWR

**Group**

Power

**Syntax**

- `POWer:POWer<x>:PSRR:PPD <NR3>`
- `POWer:POWer<x>:PSRR:PPD?`
Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR3> is the PPD value for the measurement, in the range of 10 to 100 points.

Examples
POWer:POWer3:PSRR:PPD 30 sets the PPD for Power measurement 3 to be 30 points.

POWer:POWer5:PSRR:PPD? might return 20, indicating that the PPD value for power measurement 5 is 20 points.

POWer:POWer<x>:PSRR:STARTFREQuency

This command sets or queries the start frequency value for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions
Requires option 5-PWR or 6-PWR

Group
Power

Syntax
POWer:POWer<x>:PSRR:STARTFREQuency <NR3>
POWer:POWer<x>:PSRR:STARTFREQuency?

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR3> is the starting frequency for the measurement, in the range of 10 Hz to 50 MHz.

Examples
POWer:POWer8:PSRR:STARTFREQuency 12 sets the starting frequency for power measurement 8 to be 12 Hz.

POWer:POWer3:PSRR:STARTFREQuency? might return 2400, indicating that the starting frequency for power measurement 3 is 2400 Hz.

POWer:POWer<x>:PSRR:STOPFREQuency

This command sets or queries the stop frequency value for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions
Requires option 5-PWR or 6-PWR
Commands listed in alphabetical order

Group: Power

**Syntax**

POWer:POWer<x>:PSRR:STOPFREQuency <NR3>
POWer:POWer<x>:PSRR:STOPFREQuency?

**Arguments**

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR3> is the stop frequency for the measurement, in the range of 10 Hz to 50 MHz.

**Examples**

POWer:POWer3:PSRR:STOPFREQuency 120 sets the stop frequency for Power measurement 8 to be 120 Hz.

POWer:POWer5:PSRR:STOPFREQuency? might return 2000, indicating that the stop frequency for power measurement 5 is 2000 Hz.

**POWer:POWer<x>:RDSON:DEVICEType**

This command sets or queries the device type for the power drain source on resistance measurement for RDSon measurement of the specified power measurement number.

**Conditions**

Requires option PWR or PS2.

**Group:** Power

**Syntax**

POWer:POWer<x>:RDSON:DEVICEType {SWITCHING | PNJUNCTION}
POWer:POWer<x>:RDSON:DEVICEType?

**Arguments**

Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

SWITCHING sets the Device Type to a switching device (v/i).

PNJUNCTION sets the Device Type to a PN Junction device (dv/di).

**Examples**

POWer:POWer3:RDSON:DEVICEType PNJUNCTION sets the RDSon power measurement number 3 to measure a PN junction device.

POWer:POWer1:RDSON:DEVICEType? might return SWITCHING, indicating that RDSon power measurement number 1 is set to measure a switching device.
**POWer:POWer<x>:RDSON:ISOURce**

This command sets or queries the current source for RDSon measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```plaintext
POWer:POWer<x>:RDSON:ISOURce {CH<x>|MATH<x>|REF<x>}  
POWer:POWer<x>:RDSON:ISOURce?
```

**Related Commands**

POWer:POWer<x>:RDSON:VSOURce

**Arguments**

- **Power<x>** is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.
- **CH<x>** sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.
- **MATH<x>** = A math waveform specifier; <x> is ≥1.
- **REF<x>** = A reference waveform specifier; <x> is ≥1.

**Examples**

- `POWer:POWer1:RDSON:ISOURce REF2` sets RDSon power measurement number 1 to use Reference waveform 2 as the current signal source.
- `POWer:POWer1:RDSON:ISOURce?` might return `CH1`, indicating that channel 1 is the current signal source for the RDSon power measurement number 1.

**POWer:POWer<x>:RDSON:VSOURce**

This command sets or queries the voltage source for RDSon measurement of the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```plaintext
POWer:POWer<x>:RDSON:VSOURce {CH<x>|MATH<x>|REF<x>}  
POWer:POWer<x>:RDSON:VSOURce?
```

For more details, refer to the MSO4, MSO5, MSO8, MSO8LP, and MSO64 Programmer manual.
Related Commands
POWer:POWer<x>:RDSON:ISOURce

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples
POWer:POWer2:RDSON:VSOURce CH7 sets RDSon power measurement number 2 to use channel 7 as the voltage source.

POWer:POWer1:RDSON:VSOURce? might return CH7, indicating that channel 1 is the voltage signal source for RDSon power measurement number 1.

POWer:POWer<x>:REFLevels:ABSolute:FALLHigh

This command sets or queries the falling edge for high reference level in absolute units for the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:REFLevels:ABSolute:FALLHigh <NR1>
POWer:POWer<x>:REFLevels:ABSolute:FALLHigh?

Related Commands
POWer:POWer<x>:REFLevels:METHOD
POWer:POWer<x>:REFLevels:ABSolute:TYPE

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR1> ranges from –40000 to 40000

Examples
POWer:POWer1:REFLevels:ABSolute:FALLHigh 1 sets the high reference level for falling edge as 1 volt.
POWer:POWer<x>:REFLevels:ABSolute:FALLLow

This command sets or queries the falling edge for low reference level in absolute units for the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:REFLevels:ABSolute:FALLLow <NR1>
POWer:POWer<x>:REFLevels:ABSolute:FALLLow?

Related Commands
POWer:POWer<x>:REFLevels:METHod
POWer:POWer<x>:REFLevels:ABSolute:TYPE

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR1> ranges from –40000 to 40000

Examples
POWer:POWer1:REFLevels:ABSolute:FALLLow –1 sets the low reference level for falling edge as –1 volt.

POWer:POWer<x>:REFLevels:ABSolute:FALLMid

This command sets or queries the falling edge for mid reference level in absolute units for the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:REFLevels:ABSolute:FALLMid <NR1>
POWer:POWer<x>:REFLevels:ABSolute:FALLMid?

Related Commands
POWer:POWer<x>:REFLevels:METHod
POWer:POWer<x>:REFLevels:ABSolute:TYPE
**Arguments**

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NRI> ranges from -40000 to 40000

**Examples**

`POWER:POWER1:REFLevels:ABSolute:FALLMid 0` sets the mid reference level for falling edge as 0 volt.

**POWER:POWER<x>:REFLevels:ABSolute:HYSTeresis**

This command sets or queries the absolute hysteresis value for the specified power measurement number.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

`POWER:POWER<x>:REFLevels:ABSolute:HYSTeresis <NRI>`

`POWER:POWER<x>:REFLevels:ABSolute:HYSTeresis?`

**Related Commands**

`POWER:POWER<x>:REFLevels:METHOD`
Commands listed in alphabetical order

Syntax

POWer:POWer<x>:REFLevels:ABSolute:RISEHigh <NR1>
POWer:POWer<x>:REFLevels:ABSolute:RISEHigh?

Related Commands

POWer:POWer<x>:REFLevels:METHod
POWer:POWer<x>:REFLevels:ABSolute:TYPE

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR1> ranges from –40000 to 40000

Examples

POWer:POWer1:REFLevels:ABSolute:RISEHigh 1 sets the high reference level for rising edge as 1 volt.

POWer:POWer<x>:REFLevels:ABSolute:RISELow

This command sets or queries the rising edge for low reference level in absolute units for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:REFLevels:ABSolute:RISELow <NR1>
POWer:POWer<x>:REFLevels:ABSolute:RISELow?

Related Commands

POWer:POWer<x>:REFLevels:METHod
POWer:POWer<x>:REFLevels:ABSolute:TYPE

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR1> ranges from –40000 to 40000.

Examples

POWer:POWer1:REFLevels:ABSolute:RISELow –1 sets the low reference level for rising edge as –1 volt.
**POWer:POWer<x>:REFLevels:ABSolute:RISEMid**

This command sets or queries the rising edge for mid reference level in absolute units for the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:REFLevels:ABSolute:RISEMid <NR1>
POWer:POWer<x>:REFLevels:ABSolute:RISEMid?
```

**Related Commands**

- `POWer:POWer<x>:REFLevels:METHod`
- `POWer:POWer<x>:REFLevels:ABSolute:TYPE`

**Arguments**

- `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

- `<NR1>` ranges from –40000 to 40000.

**Examples**

```
POWer:POWer1:REFLevels:ABSolute:RISEMid 0
```
sets the mid reference level for rising edge as 0 volt.

---

**POWer:POWer<x>:REFLevels:ABSolute:TYPE**

This command sets or queries the type of measurement levels when reference level is set to absolute for the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:REFLevels:ABSolute:TYPE {SAME | UNIQue}
POWer:POWer<x>:REFLevels:ABSolute:TYPE?
```

**Related Commands**

- `POWer:POWer<x>:REFLevels:METHod`

---
Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

SAME: select when the rising edge and falling edge reference levels are same.

UNIQUE: select when the rising edge and falling edge reference levels are different.

Examples

POWER:POWER1:REFLevels:ABSolute:TYPE UNIQUE sets the type of measurement levels as unique for the specified power measurement badge.

POWER:POWER<x>:REFLevels:BASETop

This command sets or queries the reference level base top method for the specified power measurement number. The power measurement number is specified by <x>.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWER:POWER<x>:REFLevels:BASETop {AUTO | MINMax | MEANhistogram | MODEhistogram | EYEhistogram}

POWER:POWER<x>:REFLevels:BASETop?

Related Commands

POWER:POWER<x>:REFLevels:METHod

Examples

POWER:POWER1:REFLevels:BASETop AUTO sets the reference level base top method as auto for the power measurement badge 1.

POWER:POWER<x>:REFLevels:METHod

This command sets or queries the method to configure reference level values for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWER:POWER<x>:REFLevels:METHod {PERCent | ABSolute}

POWER:POWER<x>:REFLevels:METHod?
Arguments

Power<xx> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

PERCent sets the power measurement to use absolute values to configure reference level values.

ABSolute sets the power measurement to use percentage to configure reference level values.

Examples

POWer:POWer3:REFLevels:METHod PERCent sets power measurement 3 to use percentage to configure reference level values.

POWer:POWer<xx>:REFLevels:PERCent:FALLHigh

This command sets or queries the falling edge for high reference level in percentage for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<xx>:REFLevels:PERCent:FALLHigh <NR1>
POWer:POWer<xx>:REFLevels:PERCent:FALLHigh?

Related Commands

POWer:POWer<xx>:REFLevels:METHod
POWer:POWer<xx>:REFLevels:PERCent:TYPE

Arguments

Power<xx> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR1> ranges from 1 to 99.

Examples

POWer:POWer1:REFLevels:PERCent:FALLHigh 1 sets the high reference level for falling edge as 1 percentage.

POWer:POWer<xx>:REFLevels:PERCent:FALLLow

This command sets or queries the falling edge for low reference level in percentage for the specified power measurement number.
Commands listed in alphabetical order

POWer:POWer<x>:REFLevels:PERCent:FALLow

Syntax
POWer:POWer<x>:REFLevels:PERCent:FALLow <NR1>
POWer:POWer<x>:REFLevels:PERCent:FALLow?

Related Commands
POWer:POWer<x>:REFLevels:METHod
POWer:POWer<x>:REFLevels:PERCent:TYPE

Arguments
power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
<NR1> ranges from 1 to 99.

Examples
POWer:POWer1:REFLevels:PERCent:FALLow 0 sets the low reference level for falling edge as 0 percentage.

POWer:POWer<x>:REFLevels:PERCent:FALLMid

This command sets or queries the falling edge for mid reference level in percentage for the specified power measurement number.

Syntax
POWer:POWer<x>:REFLevels:PERCent:FALLMid <NR1>
POWer:POWer<x>:REFLevels:PERCent:FALLMid?

Related Commands
POWer:POWer<x>:REFLevels:METHod
POWer:POWer<x>:REFLevels:PERCent:TYPE

Arguments
power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
<NR1> ranges from 1 to 99.
Examples: `POWer:POWer1:REFLevels:PERCent:FALLMid 50` sets the mid reference level for falling edge as 50 percentage.

**POWer:POWer<x>:REFLevels:PERCent:HYSTeresis**

This command sets or queries the hysteresis in percentage for the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:REFLevels:PERCent:HYSTeresis <NR1>
POWer:POWer<x>:REFLevels:PERCent:HYSTeresis?
```

**Related Commands**
`POWer:POWer<x>:REFLevels:METHOD`

**Arguments**
`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from 1 to 99.

**Examples**

```
POWer:POWer1:REFLevels:PERCent:HYSTeresis 25
```

sets the hysteresis as 25 percentage.

**POWer:POWer<x>:REFLevels:PERCent:RISEHigh**

This command sets or queries the rising edge for high reference level in percentage for the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:REFLevels:PERCent:RISEHigh <NR1>
POWer:POWer<x>:REFLevels:PERCent:RISEHigh?
```

**Arguments**

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from 1 to 99.

**Examples**

```
POWer:POWer1:REFLevels:PERCent:RISEHigh 25
```

sets the hysteresis as 25 percentage.
Related Commands

POWer:POWer<x>:REFLevels:METHOD
POWer:POWer<x>:REFLevels:PERCent:TYPE

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<N1> ranges from 1 to 99.

Examples

POWer:POWer1:REFLevels:PERCent:RISEHigh 85 sets the high reference level for rising edge as 85 percentage.

POWer:POWer<x>:REFLevels:PERCent:RISELow

This command sets or queries the rising edge for low reference level in percentage for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:REFLevels:PERCent:RISELow <N1>
POWer:POWer<x>:REFLevels:PERCent:RISELow?

Related Commands

POWer:POWer<x>:REFLevels:METHOD
POWer:POWer<x>:REFLevels:PERCent:TYPE

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<N1> ranges from 1 to 99.

Examples

POWer:POWer1:REFLevels:PERCent:RISELow 10 sets the low reference level for rising edge as 10 percentage.

POWer:POWer<x>:REFLevels:PERCent:RISEMId

This command sets or queries the rising edge for mid reference level in percentage for the specified power measurement number. The power measurement number is specified by <x>.
Commands listed in alphabetical order

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
```
POWER:POWer<x>:REFLevels:PERCent:RISEMid <NR1>
POWER:POWer<x>:REFLevels:PERCent:RISEMid?
```

**Related Commands**
- POWER:POWer<x>:REFLevels:METHod
- POWER:POWer<x>:REFLevels:PERCent:TYPE

**Arguments**
- `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
- `<NR1>` ranges from 1 to 99.

**Examples**
```
POWER:POWer1:REFLevels:PERCent:RISEMid 55
```

sets the mid reference level for rising edge as 55 percentage.

---

**POWER:POWer<x>:REFLevels:PERCent:TYPE**

This command sets or queries the reference levels for the specified power measurement number.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
```
POWER:POWer<x>:REFLevels:PERCent:TYPE {TENNinety | TWENtyeighty | CUSTom}
POWER:POWer<x>:REFLevels:PERCent:TYPE?
```

**Related Commands**
- POWER:POWer<x>:REFLevels:METHod
- POWER:POWer<x>:REFLevels:PERCent:FALLHigh
- POWER:POWer<x>:REFLevels:PERCent:FALLLow
- POWER:POWer<x>:REFLevels:PERCent:FALLMid
- POWER:POWer<x>:REFLevels:PERCent:RISEHigh
- POWER:POWer<x>:REFLevels:PERCent:RISELow
**POWer:POWer<x>:REFlEvels:PERCenT:RISEMid**

**Arguments**

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

**TENNinety** to set the low reference levels as 10% and high reference levels as 90%

**TWENTyeighty** to set the low reference levels as 20% and high reference levels as 80%

**CUSTom** to set the custom low, high, and mid reference levels for rising and falling edges

**Examples**

POWer:POWer1:REFlEvels:PERCenT:TYPE TENNinety sets the low reference levels as 10% and high reference levels as 90% for the power measurement badge 1.

**POWer:POWer<x>:RESUlts:ALLAcqs:MAXimum?** *(Query Only)*

This command queries the maximum value of all acquisitions for the measurement parameter in the specified power measurement number. The power measurement number is specified by <x>.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

POWer:POWer<x>:RESUlts:ALLAcqs:MAXimum? {InputPwr|Output1Pwr|Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|Hc|Hmax|IRipple|DeltaB|DeltaH|Permeability|RDS|TRUEPwr|APPPwr|REPwr|PWRFACtor|PHASE|PWRFREQ|ICFACtor|VCFACtor|IRMS|VRMS|TONENRG|TOFFENRG|TOFFLOSS|CONDENRG|CONLOSS|TTLLOSS|TTLENRG|DVBYDT|DIBYDT|SOAHITS|CTR|LRI|LRIPPK|SWLRIPPK|SWLRIPPK|PRIOD|FREQ|PDU|NPULSE|NPPULSE|AMPL|PKPK|HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|MAXPSRR|MAXPSRR|MINPSRR|MINPSRRFREQ}
NOTE. Above entries are <Qstring> entries, and must be entered in enclosing quotes.

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<Qstring> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

For the Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dl by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.
"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement.

Examples

POWer:POWer4:RESUlts:ALLAcqs:MAXimum? "PKPK" might return 28.56, indicating the maximum value of the Peak-to-Peak measurement for power measurement 4.

POWer:POWer<x>:RESUlts:ALLAcqs:MEAN? (Query Only)

This command queries the mean value of all acquisitions for the measurement parameter in the specified power measurement number <x>.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:RESUlts:ALLAcqs:MEAN?
{InputPwr|Output1Pwr|Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|Hc|Hmax|IRipple|DeltaB|DeltaH|Permeability|RDS|TRUEPWR|APPWR|REPWR|PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR|IRMS|VRMS|TONENRG|TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|
Arguments

Power<\(\times\)> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<\texttt{QString}> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <\texttt{QString}> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFATOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the d\(V\) by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.
"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement.

**Examples**

```
POWer:POWer2:RESUlts:ALLAcqs:MEAN? "AMPL" might return 57.45, indicating the mean value of the Amplitude measurement for all acquisitions of power measurement 2.
```

**POWer:POWer<x>:RESUlts:ALLAcqs:MINimum? (Query Only)**

This command queries the minimum value of all acquisitions for the measurement parameter of the specified power measurement <x>.

**Conditions**

Requires option PWR or PS2.

**Group**

Power
Syntax

`POWeRx:RESUtx:ALLAcq:MINimum?` {InputPwr|Output1Pwr|Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|Hc|Hmax|IRipple|DeltaB|DeltaH|Permeability|RDS|TRUEPWR|APPPWR|REPWx|PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR|IRMS|VRMS|TONENRG|TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|TTLLOSS|TTLENRG|DVBYx|DIBYx|SOAHITSCNT|LRIIPRMS|LRIIPPPKPK|SWRIPRMS|SWRIPPKPK|PRIOD|FREQ|PDUTY|NDUTY|PPULSE|NPULSE|AMPL|PKPK|HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|MAXPSRR|MXPSSRFREQ|MINPSRR|MINPSRRFREQ}

NOTE. Above entries are `<QString>` entries, and must be entered in enclosing quotes.

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<QString>` = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid `<QString>` arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVS Integral V measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWx", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYx" is the parameter for the dV by dt measurement.
"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHTSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement.

Examples

POWer:POWer5:RESUlts:ALLAcqs:MINimum? "FREQ" might return 5.1307829019093E6, indicating the minimum frequency measurement for all acquisitions of power measurement 5.

POWer:POWer<x>:RESUlts:ALLAcqs:PK2PK? (Query Only)

This command queries the peak-to-peak value of all acquisitions for the measurement parameter in the specified power measurement number.
### Conditions
Requires option PWR or PS2.

### Group
Power

### Syntax
```
POWer:POWer<x>:RESUlt:ALLAcqs:PK2PK? {InputPwr|Output1Pwr|
Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|
TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|
Br|Hc|Hmax|IRipple|DeltaB|DeltaH|
Permeability|RDS|TRUEPWR|APPPWR|REPWR|
PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR|
IRMS|VRMS|TONENRG|TONLOSS|TOFFENRG|
TOFFLOSS|CONDENRG|CONDLOSS|TTLLOSS|TTLENRG|
DVBYDT|DIBYDT|SOAHITSCNT|LRIPRMS|LRIPPKPK|
SWRIPRMS|SWRPKPK|PRIOD|FREQ|PDUTY|
NDUTY|PPULSE|NPULSE|AMPL|PKPK|
HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|
OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|
MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

**NOTE.** Above entries are `<QString>` entries, and must be entered in enclosing quotes.

### Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <QString> arguments are:

- For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".  
- For Inductance measurement, the parameter is "INDUCT".  
- For IVSIntegralV measurement, the parameter is "IVSINTV".  
- For Magnetic Loss measurement, the parameter is "MAGLOSS".  
- For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".  
- For RDSon measurement, the parameter is "RDS".  
- For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".  

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer

2-683
For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLENRG", "TTTLLOSS".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak-Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement.

Examples

POWer:POWer<x>:RESUIts:ALLAcqs:POPUlation? (Query Only)

This command queries the population (number of complete cycles) of all acquisitions for the measurement parameter in the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:RESUIts:ALLAcqs:POPUlation? {InputPwr|Output1Pwr|Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|Hmax|IRipple|DeltaB|DeltaH|Permeability|RDS|TRUEPWR|APPPWR|REPW|PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFATOR|IRMS|VRMS|TONENRG|TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|TTLLOSS|TTLENRG|DBYDT|DIBYDT|SOCMNT|LRIPMS|LRIPPKPK|SWPRMS|SWRIPPKPK|PRIO|FREQ|PDUTY|NDUTY|PPULSE|NPULSE|AMPL|PKPK|HIGH|LOW|MAX|MIN|INRUSH|CAPNCE|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}

NOTE. Above entries are <QString> entries, and must be entered in enclosing quotes.

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <QString> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".
For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"D\text{V}B\text{YDT}\"" is the parameter for the dV by dt measurement.

"D\text{I}B\text{YDT}\" is the parameter for the dI by dt measurement.

"SOAHITSCNT\" is the parameter for the SOA measurement.

"LRIPRMS\" and "LRIPPKPK\" are the parameters for the Line Ripple measurement.

"SWRIPRMS\" and "SWRIPPKPK\" are the parameters for the Switching Ripple measurement.

"PRIOD\" is the parameter for the Cycle Period measurement.

"FREQ\" is the parameter for the Cycle Frequency measurement.

"PDUTY\" is the parameter for the Positive Duty Cycle measurement.

"NDUTY\" is the parameter for the Negative Duty Cycle measurement.

"PPULSE\" is the parameter for the Positive Pulse Width measurement.

"NPULSE\" is the parameter for the Negative Pulse Width measurement.

"AMPL\" is the parameter for the Cycle Amplitude measurement.

"PKPK\" is the parameter for the Cycle Peak–Peak measurement.

"HIGH\" is the parameter for the Cycle Top measurement.

"LOW\" is the parameter for the Cycle Base measurement.

"Max\" is the parameter for the Cycle Max measurement.

"MIN\" is the parameter for the Cycle Min measurement.

"INRUSH\" is the parameter for the Inrush Current measurement.

"CAPACITANCE\" is the parameter for the Input Capacitance measurement.

"OUTPUT1\" - "OUTPUT7\" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.
"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement.

Examples

```
POWer:POWer1:RESUIts:ALLAcqs:POPUlation? "CondEnrg" might return 4.91E-3, indicating the population (number of complete cycles) of conduction energy for all acquisitions.
```

**POWer:POWer<x>:RESUIts:ALLAcqs:STDDev? (Query Only)**

This command queries the standard deviation value of all acquisitions for the measurement parameter in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUIts:ALLAcqs:STDDev? {InputPwr|Output1Pwr|Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|Hc|Hmax|IRipple|DeltaB|DeltaH|Permeability|RDS|TRUEPWR|APPPWR|REPWR|PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR|IRMS|VRMS|TONENRG|TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|TTLLOSS|TTLENRG|DVBYDT|DYDT|SOAHITSCTN|LRIPRMS|LRIPPKPK|SWRIPRMS|SWRIPPKPK|PRIOD|FREQ|PDUTY|NDUTY|PPULSE|NPULSE|AMPL|PKPK|HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

**NOTE.** Above entries are `<QSTring>` entries, and must be entered in enclosing quotes.

Arguments

<Power<x> > is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:
For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFATOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLLOSS".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.
"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement.

Examples


POWer:POWer<x>:RESUlts:CURRentacq:F1MAG? (Query Only)

This command queries the first harmonics magnitude value for the specified power measurement number.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:RESUlts:CURRentacq:F1MAG? "harmonics"

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples
POWer:POWER1:RESUlts:CURRentacq:F1MAG? "harmonics" might return 1.4151834770090, indicating the value of the first harmonics magnitude for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:F3MAG? (Query Only)

This command queries the third harmonics magnitude value for the specified power measurement number.

Conditions
Requires option PWR or PS2.
Commands listed in alphabetical order

**Group**  
Power

**Syntax**  
POWer:POWer<x>:RESUlts:CURRentacq:F3MAG? "harmonics"

**Arguments**  
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

**Examples**  
POWer:POWer1:RESUlts:CURRentacq:F3MAG? "harmonics" might return 234.0187140104806E-6, indicating the value of third harmonics magnitude for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:FREQUENCY? (Query Only)

This command queries the fundamental frequency for the specified power measurement number.

**Conditions**  
Requires option PWR or PS2.

**Group**  
Power

**Syntax**  
POWer:POWer<x>:RESUlts:CURRentacq:FREQUENCY? "harmonics"

**Arguments**  
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

**Examples**  
POWer:POWer1:RESUlts:CURRentacq:FREQUENCY? "harmonics" might return 100.0067656931537E+3, indicating the fundamental frequency for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:IRMS? (Query Only)

This command queries the RMS current value for the specified power measurement number.

**Conditions**  
Requires option PWR or PS2.

**Group**  
Power
POWer:POWeRx:RESUIt:sCURRentacq:IRMS? "harmonics"

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples

POWer:POWeRx:RESUIt:sCURRentacq:IRMS? "harmonics" might return 1.414980733491, indicating the RMS current value for the power measurement badge 1.

POWer:POWeRx:RESUIt:sCURRentacq:MAXimum? (Query Only)

This command queries the maximum value of the current acquisition for the measurement parameter in the specified power measurement number.

Syntax

POWer:POWeRx:RESUIt:sCURRentacq:MAXimum? {InputPwr|Output1Pwr|Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|Hc|Hmax|IRipple|DeltaB|DeltaH|Permeability|RDS|TRUEPWR|APPPWR|REPWR|PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR|IRMS|VRMS|TONENRG|TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|TTLLOSS|TTLENRG|DVBYDT|DIBYDT|SOAHITSCNT|LRIPRMS|LRIPPKPK|SWRIPRMS|SWRIPPKPK|PRIOD|FREQ|PDUTY|NDUTY|PPULSE|NPULSE|AMPL|PKPK|HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<Qstring> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:
For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFATOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLNREG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.
"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement.

Examples

POWER:POWER1:RESULTS:CURRENTACQ:MAdimum? "TONEnrg" might return 32.8, indicating the maximum value of Ton Energy for the current acquisition.

POWeR:POWeRx:RESUIts:CURREnTACQ:MEAn? (Query Only)

This command queries the mean value of the current acquisition for the measurement parameter of the specified power measurement <x>.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWER:POWER<x>::RESULTS:CURRENTACQ:MEAN? {InputPwr|Output1Pwr|Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|Hc|Hmax|IRipple|DeltaB|DeltaH|Permeability|RDS|TRUEPWR|APPPWR|REPPWR|PWRFACTOR|PHASE|PwRFREQ|ICFACTOR|VCFACTOR|IRMS|VRMS|TONENRG|TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|TTLLOSS|TTLENRG|DVBYDT|DIBYDT|SOAHITSCNT|LRIPRMS|LRIPPKPK|SWRIPRMS|SWRIPPKPK|PRIOD|FREQ|PDUTY|NDUTY|PPULSE|NPULSE|AMPL|PKPK|HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}

NOTE. Above entries are <QSTring> entries, and must be entered in enclosing quotes.

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
<Qstring> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.
"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement.

Examples

POWer:POWer2:RESUlts:CURRentacq:MEAN? "TruePwr" might return 42.6097255943E–2, indicating the mean value of true power for the current acquisition of power measurement 2.

POWer:POWer<x>:RESUlts:CURRentacq:MINimum? (Query Only)

This command queries the minimum value of the current acquisition for the measurement parameter in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:RESUlts:CURRentacq:MINimum? {InputPwr|Output1Pwr|Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Peak|Br|Hc|Hmax|IRipple|DeltaT|DeltaH|Permeability|RDS|TRUEPwr|APPwr|REPPwr|PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR|IRMS|VRMS|TONENRG|TONLOSS|TONFENRG|TONLOSS|CONDENRG|CONDLOSS|TTLOSS|TTLENRG|DBYD|DBYDT|SOAHITSCNT|LRIPRMS|LRIPPKP|SWIPRMS|SWIPPPKPK|PRIOD|FREQ|PDTY|NDUTY|PPULSE|NPULSE|AMPL|PKPK|HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}

NOTE. Above entries are <QSTring> entries, and must be entered in enclosing quotes.
Commands listed in alphabetical order

Arguments

Power <x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <QString> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPFWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFATOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.
"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement.

Examples

POWer:POWer1:RESUlts:CURRentacq:MINimum? "TruePwr" might return 4.17829019093E–9, indicating the minimum value of true power for the current acquisition.

POWer:POWer<x>:RESUlts:CURRentacq:PK2PK? (Query Only)

This command queries the peak-to-peak value of the current acquisition for the measurement parameter in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:RESUlts:CURRentacq:PK2PK? {InputPwr|Output1Pwr|Output2Pwr|Output3Pwr|Output4Pwr|Output5Pwr|Output6Pwr|Output7Pwr|Efficiency1|Efficiency2|Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|Hc|Hmax|IRipple|DeltaB|DeltaH|Permeability|RDS|TRUEPWR|APPPWR|REPWR|PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR|IRMS|VRMS|TONENRG|TONLOSS|TOFFNRG|TOFFLOSS|CONDENRG|CONDLOSS|TTLENRG|DTBYDT|DIbyDT|SOAHITSCNT|LRIPPKPK|LRIPPRMS|SWIRPPKPK|SWIRPPRMS|SWRIPPKPK|SWRIPPRMS|PRIOD|FREQ|PDUTY|NDUTY|PPULSE|NPULSE|AMPL|PKPK|HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
NOTE. Above entries are \texttt{QString} entries, and must be entered in enclosing quotes.

Arguments

Power\texttt{<x>} is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

\texttt{QString} = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid \texttt{QString} arguments are:

- For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".
- For Inductance measurement, the parameter is "INDUCT".
- For IVSIntegralV measurement, the parameter is "IVSINTV".
- For Magnetic Loss measurement, the parameter is "MAGLOSS".
- For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".
- For RDSon measurement, the parameter is "RDS".
- For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFATOR", "IRMS", "VRMS".
- For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".
- "D\textsubscript{VBYDT}" is the parameter for the d\textsubscript{V} by dt measurement.
- "D\textsubscript{IBYDT}" is the parameter for the d\textsubscript{I} by dt measurement.
- "SOAHITSCNT" is the parameter for the SOA measurement.
- "LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.
- "SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.
- "PRIOD" is the parameter for the Cycle Period measurement.
- "FREQ" is the parameter for the Cycle Frequency measurement.
- "PDUTY" is the parameter for the Positive Duty Cycle measurement.
- "NDUTY" is the parameter for the Negative Duty Cycle measurement.
- "PPULSE" is the parameter for the Positive Pulse Width measurement.
"NPULSE" is the parameter for the Negative Pulse Width measurement.
"AMPL" is the parameter for the Cycle Amplitude measurement.
"PKPK" is the parameter for the Cycle Peak–Peak measurement.
"HIGH" is the parameter for the Cycle Top measurement.
"LOW" is the parameter for the Cycle Base measurement.
"Max" is the parameter for the Cycle Max measurement.
"MIN" is the parameter for the Cycle Min measurement.
"INRUSH" is the parameter for the Inrush Current measurement.
"CAPACITANCE" is the parameter for the Input Capacitance measurement.
"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.
"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.
"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement.

Examples


POWer:POWer<x>:RESUlts:CURRentacq:POHCL? (Query Only)

This command queries the limit of partial odd harmonic current for the specified power measurement number. The power measurement number is specified by <x>.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWER:POWer<x>:RESUlts:CURRentacq:POHCL? "harmonics"

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples
POWeR:POWeRx:RESUlts:CURRentacq:POHCM? (Query Only)

This command queries the measured value of partial odd harmonic current for the specified power measurement number. The power measurement number is specified by <x>.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWeR:POWeRx:RESUlts:CURRentacq:POHCM? "harmonics"

Examples
POWeR:POWeR1:RESUlts:CURRentacq:POHCM? "harmonics" might return 515.4261778200E-6, indicating the measured value of partial odd harmonic current for the power measurement badge 1.

POWeR:POWeRx:RESUlts:CURRentacq:POHCS? (Query Only)

This command queries the status of partial odd harmonic current for the specified power measurement number. The power measurement number is specified by x.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWeR:POWeRx:RESUlts:CURRentacq:POHCS? "harmonics"

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples

POWeR:POWeRx:RESUlts:CURRentacq:POPUlation? (Query Only)

This command queries the population (number of complete cycles) of the current acquisition for the measurement parameter in the specified power measurement number. The power measurement number is specified by <x>.
 Commands listed in alphabetical order

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
```
POWer:POWer<x>:RESUlts:CURRentacq:POPUlation? {InputPwr|Output1Pwr|Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|Hc|Hmax|IRipple|DeltaB|DeltaH|Permeability|RDS|TRUEPWR|APPPWR|REPWR|PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFATOR|IRMS|VRMS|TONENRG|TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|TTLLOSS|TTLENRG|DVBYDT|DIBYDT|SOAHITSCNT|LRIPRMS|LRIPPKPK|SWRIPMS|SWRIPPKPK|PRIOD|FREQ|PDUTY|NDUTY|PPULSE|NPULSE|AMPL|PKPK|HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

**NOTE.** Above entries are <QString> entries, and must be entered in enclosing quotes.

**Arguments**
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <QString> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFATOR", "IRMS", "VRMS".

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer 2-701
For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement.

**Examples**

```
```
POWer:POWer<x>:RESUlts:CURRentacq:RMS? (Query Only)

This command queries the RMS value of the source selected for the specified power measurement number. The power measurement number is specified by <x>.

**Conditions**  Requires option PWR or PS2.

**Group**  Power

**Syntax**  `POWer:POWer<x>:RESUlts:CURRentacq:RMS? "harmonics"`

**Arguments**  `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

**Examples**  `POWer:POWer1:RESUlts:CURRentacq:RMS? "harmonics"` might return 1.414320437461, indicating the RMS value of the source selected for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:STATUS? (Query Only)

This command queries the status of the measurement for the specified power measurement number. The power measurement number is specified by <x>.

**Conditions**  Requires option PWR or PS2.

**Group**  Power

**Syntax**  `POWer:POWer<x>:RESUlts:CURRentacq:STATUS? "harmonics"`

**Arguments**  `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

**Examples**  `POWer:POWer1:RESUlts:CURRentacq:STATUS? "harmonics"` might return Pass, indicating that the measurement has passed according to the standard for power measurement 1.
POWer:POWer<x>:RESUlts:CURRentacq:STDDev? (Query Only)

This command queries the standard deviation value of the current acquisition for the measurement parameter in the specified power measurement number. The power measurement number is specified by <x>.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:RESUlts:CURRentacq:STDDev?
{InputPwr|Output1Pwr|
Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|
TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|Hc|Hmax|IRipple|
DeltaB|DeltaH|Permeability|RDS|TRUEPwr|APPWrr|REPWR|PWRFACTOR|
PHASE|PWRFREQ|ICFACTOR|VCFACTOR|IRMS|VRMS|TONENRG|TONLOSS|
TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|TTLENRG|TTLLOSS|DVBYDT|
DIBYDT|SOAHITSCTNT|LRIPRMS|LRIPPKPK|SWIRPRMS|SWIPPKPK|PRIOD|
FREQ|PDTY|NDUTY|PPULSE|NPULSE|AMPL|PKPK|HIGH|LOW|MAX|MIN|
INRUSH|CAPACITANCE|OUTPUT1|
OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
GAINCROSsoVERFREQ|PHASECROSSoVERFREQ|GM|PM|
MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}

NOTE. Above entries are <QString> entries, and must be entered in enclosing quotes.

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <QString> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

2-704 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFATOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dl by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement.
Examples

\texttt{POWer:POWer2:RESUIts:CURRentacq:STDDev?} "TONLoss" might return 1.31E+2, indicating the standard deviation value of Ton Energy for current acquisition of power measurement 2.

\textbf{POWer:POWer<x>:RESUIts:CURRentacq:THDF? (Query Only)}

This command queries the total harmonic distortion (fundamental) value for the specified power measurement number. The power measurement number is specified by \texttt{<x>}.

\textbf{Conditions} Requires option PWR or PS2.

\textbf{Group} Power

\textbf{Syntax} \texttt{POWer:POWer<x>:RESUIts:CURRentacq:THDF?} "harmonics"

\textbf{Arguments} \texttt{Power<x>} is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

\textbf{Examples} \texttt{POWer:POWer1:RESUIts:CURRentacq:THDF?} "harmonics" might return 96.984696670887E-3, indicating the value of total harmonic distortion (fundamental) for power measurement 1.

\textbf{POWer:POWer<x>:RESUIts:CURRentacq:THDR? (Query Only)}

This command queries the total harmonic distortion (RMS) value for the specified power measurement number. The power measurement number is specified by \texttt{<x>}.

\textbf{Conditions} Requires option PWR or PS2.

\textbf{Group} Power

\textbf{Syntax} \texttt{POWer:POWer<x>:RESUIts:CURRentacq:THDR?} "harmonics"

\textbf{Arguments} \texttt{Power<x>} is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:THDR? "harmonics" might return 107.1384597967292E-3, indicating the value of total harmonic distortion (fundamental) for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:TRPWR? (Query Only)

This command queries the true power value for the specified power measurement number. The power measurement number is specified by <x>.

Conditions Requires option PWR or PS2.

Group Power

Syntax POwer:POwer<x>:RESUlts:CURRentacq:TRPWR? "harmonics"

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples POwer:POwer1:RESUlts:CURRentacq:TRPWR? "harmonics" might return 2.0002612633993, indicating the true power value for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:VRMS? (Query Only)

This command queries the RMS voltage value for the specified power measurement number. The power measurement number is specified by <x>.

Conditions Requires option PWR or PS2.

Group Power

Syntax POwer:POwer<x>:RESUlts:CURRentacq:VRMS? "harmonics"

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples POwer:POwer3:RESUlts:CURRentacq:VRMS? "harmonics" might return 1.4117680233354, indicating the RMS voltage value for the power measurement 3.
Commands listed in alphabetical order

**POWer:POWer<x>:SEQSETup (No Query Form)**

This command sets up the oscilloscope's horizontal, vertical, and trigger parameters to optimize for taking the specified power measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:SEQSETup Execute

**Arguments**
POWer<x> is the Power measurement identifier number. The number must be for a power measurement that requires a single sequence acquisition.

RUN sets the measurement to run an acquisition and acquire data for the specified single sequence power measurement.

**Examples**
POWer:POWer3:SEQSequence Execute runs the power measurement 3 sequence setup.

---

**POWer:POWer<x>:SEQSequence**

This command sets or queries the run state of a single sequence power measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:SEQSequence {RUN|RERUN}
POWer:POWer<x>:SEQSequence?

**Arguments**
POWer<x> is the Power measurement identifier number. The number must be for a power measurement that requires a single sequence acquisition.

RUN sets the measurement to run an acquisition and acquire data for the specified single sequence power measurement.

RERUN sets the measurement to rerun an acquisition and acquire data for the specified single sequence power measurement.
Examples

POWer:POWer3:SEQuence RUN sets power measurement 3 to run the measurement.

POWer:POWer1:SEQuence? might return POWer:POWer1:SEQuence RERUN, indicating that the current state of the power measurement is rerun.

POWer:POWer<x>:SOA:ISOURce

This command sets or queries the current source for SOA measurement in the specified power measurement number. The power measurement number is specified by <x>.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:SOA:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:SOA:ISOURce?

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels on your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples
POWer:POWer1:SOA:ISOURce CH2 sets the current source for the SOA measurement 1 as channel 2.

POWer:POWer<x>:SOA:POINT

This command sets or queries the X or Y coordinate value for an SOA mask of a specified power measurement.

Conditions
Requires option PWR or PS2.

Group
Power
Commands listed in alphabetical order

Syntax

POWer:POWer<x>:SOA:POINT <QString> <NR1>
POWer:POWer<x>:SOA:POINT <QString>?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.


<NR2> sets the specified SOA mask X or Y point value, as a floating number, in the range from –5000 to 5000.

Examples

POWer:POWer1:SOA:POINT6Y 0.125 sets the #6Y coordinate value of the SOA mask of power measurement 1 to 0.125.
POWer:POWer3:SOA:POINT12X? might return POWer:POWer3:SOA:POINT12X 1.435, indicating that the #12X coordinate value of the SOA mask of power measurement 3 is 1.435.

POWer:POWer<x>:SOA:RECAllmask

This command recalls or queries the recall mask file name in the specified power measurement number. The power measurement number is specified by <x>.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:SOA:RECAllmask
POWer:POWer<x>:SOA:RECAllmask?

Examples

POWer:POWer1:SOA:RECAllmask? might return Tek000.msk, indicating the file name of the mask that will be recalled.

POWer:POWer<x>:SOA:RECAllmask:FILEName

This command sets or queries the file name for saving SOA mask file name in the specified power measurement number. The power measurement number is specified by <x>. 
Commands listed in alphabetical order

**POWer**:POWer<x>:SOA:RECALLmask:FILENAME

This command recalls the mask file as per the name configured and at the configured path or queries the mask file name, path, and file type for the SOA measurement in the specified power measurement number. The power measurement number is specified by <x>.

**POWer**:POWer<x>:SOA:RECALLmask:FILENAME?

**POWer**:POWer<x>:SOA:SMAVemask

This command saves the mask file as per the name configured and at the configured path or queries the mask file name, path, and file type for the SOA measurement in the specified power measurement number. The power measurement number is specified by <x>.

**POWer**:POWer<x>:SOA:SMAVemask?

**POWer**:POWer<x>:SOA:SMAVemask:AUTOINCReement

This command sets or queries the state of auto-increment for saved SOA mask file names in the specified power measurement number. The power measurement number is specified by <x>.

**Related Commands**

POWer:POWer<x>:SOA:SMAVemask:FILENAME

**Examples**

POWer:POWer1:SOA:RECALLmask:FILENAME

POWer:POWer1:SOA:SMAVemask saves the mask file of SOA measurement as the configured file name at the configured path.

POWer:POWer1:SOA:SMAVemask:AUTOINCReement
Syntax

POWer:POWer<x>:SOA:SAVemask:AUTOINCrement
POWer:POWer<x>:SOA:SAVemask:AUTOINCrement?

POWer:POWer<x>:SOA:SAVemask:FILEName

This command sets or queries the mask file name for SOA measurement in the specified power measurement number. The power measurement number is specified by <x>.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax

POWer:POWer<x>:SOA:SAVemask:FILEName
POWer:POWer<x>:SOA:SAVemask:FILEName?

Examples

POWer:POWer1:SOA:SAVemask:FILEName “Tek001.msk” sets the mask file name for SOA measurement as Tek001.msk.

POWer:POWer<x>:SOA:SAVemask:FOLDer

This command sets or queries the mask file folder path for SOA measurement in the specified power measurement number. The power measurement number is specified by <x>.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax

POWer:POWer<x>:SOA:SAVemask:FOLDer
POWer:POWer<x>:SOA:SAVemask:FOLDer?

Examples

POWer:POWer1:SOA:SAVemask:FOLDer “C:” sets the mask file folder path for SOA measurement as C drive.
POWer:POWer<x>:SOA:VSOURce

This command sets or queries the voltage source for SOA measurement in the specified power measurement number. The power measurement number is specified by <x>.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:SOA:VSOURce {CH<x> | MATH<x> | REF<x>}
POWer:POWer<x>:SOA:VSOURce?

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

CH<x> = A channel specifier, <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier, <x> is ≥1.

REF<x> = A reference waveform specifier, <x> is ≥1.

Examples
POWer:POWer1:SOA:VSOURce CH1 sets the voltage source for SOA measurement as channel 1.

POWer:POWer<x>:SWITCHINGLOSS:DEVICEType

This command sets or queries the conduction calculation method for switching loss measurement in the specified power measurement number. The power measurement number is specified by <x>.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:SWITCHINGLOSS:DEVICEType {MOSFET | BJT}
POWer:POWer<x>:SWITCHINGLOSS:DEVICEType?
Commands listed in alphabetical order

**Examples**

`POWer:POWer1:SWITCHINGLOSS:DEVICeType MOSFET` sets the conduction calculation method as mosfet for switching loss measurement of the power measurement badge Power 1.

**POWer:POWer<x>:SWITCHINGLOSS:GATESOurce**

This command sets or queries the gate voltage ($V_g$) for the switching loss measurement in the specified power measurement number. The power measurement number is specified by $<x>$.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

```plaintext
POWer:POWer<x>:SWITCHINGLOSS:GATESOurce \{ CH<x> | MATH<x> | REF<x> \}
POWer:POWer<x>:SWITCHINGLOSS:GATESOurce?
```

**Arguments**

- `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
- `CH<x>` = A channel specifier; $<x>$ is 1 through 8 and is limited by the number of FlexChannels in your instrument.
- `MATH<x>` = A math waveform specifier; $<x>$ is $\geq 1$.
- `REF<x>` = A reference waveform specifier; $<x>$ is $\geq 1$.

**Examples**

`POWer:POWer1:SWITCHINGLOSS:GATESOurce MATH1` sets the gate voltage ($V_g$) source for switching loss measurement of power measurement 1 as MATH1.

`POWer:POWer2:SWITCHINGLOSS:VSOURce?` might return MATH3 indicating the gate voltage ($V_g$) for switching loss measurement of power measurement 2.

**POWer:POWer<x>:SWITCHINGLOSS:ILEVELAbs**

This command sets or queries the current level (Ton-Start & Stop) in absolute units for switching loss measurement in the specified power measurement number. The power measurement number is specified by $<x>$.

**Conditions**

Requires option PWR or PS2.
Group: Power

Syntax:

\[ \text{POWe}\text{r}:\text{POWe}<x>:\text{SWITCHINGLOSS}:\text{ILEVELAbs} \ <\text{NR1}\> \]

\[ \text{POWe}\text{r}:\text{POWe}<x>:\text{SWITCHINGLOSS}:\text{ILEVELAbs}? \]

Related Commands: POWe\text{r}:POWe<x>:SWITCHINGLOSS:LEVELUNIts

Arguments:

- Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
- <NR1> ranges from –100 to 100

Examples:

POWe\text{r}:POWe1:SWITCHINGLOSS:ILEVELAbs 1.2 sets the current level (Ton-Start & Stop) value as 1.2 for the switching loss measurement in the power measurement badge 1.

POWe\text{r}:POWe<x>:SWITCHINGLOSS:ILEVELPct

This command sets or queries the current level (Ton-Start & Stop) in percentage for switching loss measurement in the specified power measurement number. The power measurement number is specified by <x>.

Conditions: Requires option PWR or PS2.

Group: Power

Syntax:

\[ \text{POWe}\text{r}:\text{POWe}<x>:\text{SWITCHINGLOSS}:\text{ILEVELPct} \ <\text{NR1}\> \]

\[ \text{POWe}\text{r}:\text{POWe}<x>:\text{SWITCHINGLOSS}:\text{ILEVELPct}? \]

Related Commands: POWe\text{r}:POWe<x>:SWITCHINGLOSS:LEVELUNIts

Arguments:

- Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
- <NR1> ranges from 0.0001 to 90

Examples:

POWe\text{r}:POWe1:SWITCHINGLOSS:ILEVELPct 6 sets the current level (Ton-Start & Stop) value as 6 percentage for the switching loss measurement in the power measurement badge 1.
**POWer:POWer<x>:SWITCHINGLOSS:ISOURce**

This command sets or queries the current source for the switching loss measurement in the specified power measurement number. The power measurement number is specified by <x>.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
```plaintext
POWer:POWer<x>:SWITCHINGLOSS:ISOURce \{CH<x> | MATH<x> | REF<x>\}
POWer:POWer<x>:SWITCHINGLOSS:ISOURce?
```

**Arguments**
- **Power<x>** is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
- **CH<x>** = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.
- **MATH<x>** = A math waveform specifier; <x> is ≥1.
- **REF<x>** = A reference waveform specifier; <x> is ≥1.

**Examples**
- `POWer:POWer1:SWITCHINGLOSS:ISOURce CH2` sets the current source for the switching loss measurement in the specified power measurement badge as CH2.
- `POWer:POWer2:SWITCHINGLOSS:ISOURce?` might return CH1 indicating the current source for switching loss measurement of Power 2 power measurement badge.

**POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts**

This command sets or queries the level units for switching loss measurement in the specified power measurement number. The power measurement number is specified by <x>.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
```plaintext
POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts \{PERCent | ABSolute\}
POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts?
```
Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

PERCent to set the High, Mid, and Low reference levels in percentage.

ABSolute to set the High, Mid, and Low reference levels to specific signal levels.

Examples

POWer:POWer1:SWITCHINGLOSS:LEVELUNIts ABSolute sets the level units as Absolute for switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:RDSOn

This command sets or queries the RDS(on) value for switching loss measurement in the specified power measurement number. The power measurement number is specified by <x>.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:SWITCHINGLOSS:RDSOn <NR1>

POWer:POWer<x>:SWITCHINGLOSS:RDSOn?

Related Commands

POWer:POWer<x>:SWITCHINGLOSS:DEVICEType

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR1> ranges from 0 to 100

Examples

POWer:POWer1:SWITCHINGLOSS:RDSOn 2 sets the switching loss RDSOn value as 2 for switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:SWLCONFIGType

This command sets or queries the configuration type for the switching loss measurement in the specified power measurement number. The power measurement number is specified by <x>. 
Conditions: Requires option PWR or PS2.

Group: Power

Syntax:

POWer:POWer<x>:SWITCHINGLOSS:SWLCONFIGType {SMPS | PFC | FLYBACK}
POWer:POWer<x>:SWITCHINGLOSS:SWLCONFIGType?

Arguments:

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

SMPS: Select this option in case of signals without noise and ringing. The V<sub>r</sub> source is not required. Select V<sub>r</sub> source (Source 3), in case of noisy signal.

PFC: Select this option when input DUT signals are from Power Factor Correction Circuit. For this case, V<sub>r</sub> source is mandatory.

FLYBACK: Select this option when input signals are ringing. This option does not require a V<sub>r</sub> source.

Examples:

POWer:POWer1:SWITCHINGLOSS:SWLCONFIGType PFC sets the configuration type as PFC for the switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:VCESat

This command sets or queries the value for the VCE(sat) value for switching loss measurement in the specified power measurement number. The power measurement number is specified by <x>.

Conditions: Requires option PWR or PS2.

Group: Power

Syntax:

POWer:POWer<x>:SWITCHINGLOSS:VCESat <NR1>
POWer:POWer<x>:SWITCHINGLOSS:VCESat?

Related Commands:

POWer:POWer<x>:SWITCHINGLOSS:DEVICEType

Arguments:

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.
<NR1> ranges from 0.001 to 100

**Examples**

POWer:POWer1:SWITCHINGLOSS:VCESat 6 sets VCE(sat) value as 6 for switching loss measurement of the power measurement badge Power 1.

**POWer:POWer<x>:SWITCHINGLOSS:VGLevel**

This command sets or queries the gate voltage value ($V_g$ Level Ton-Start) for the switching loss measurement in the specified power measurement number. The power measurement number is specified by $<x>$.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

POWer:POWer<x>:SWITCHINGLOSS:VGLevel <NR1>

POWer:POWer<x>:SWITCHINGLOSS:VGLevel?

**Arguments**

$Power<x>$ is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

$<NR1>$ ranges from –100 to 100

**Examples**

POWer:POWer1:SWITCHINGLOSS:VGLevel 1.2 sets the gate voltage value ($V_g$ Level Ton-Start) for the switching loss measurement of the power measurement badge Power 1.

**POWer:POWer<x>:SWITCHINGLOSS:VLEVELAbs**

This command sets or queries the voltage level (Ton-Start & Stop) in absolute units for switching loss measurement in the specified power measurement number. The power measurement number is specified by $<x>$.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

POWer:POWer<x>:SWITCHINGLOSS:VLEVELAbs <NR1>

POWer:POWer<x>:SWITCHINGLOSS:VLEVELAbs?

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
**Arguments**

Power\(<x>\) is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

\(<\text{NR1}>\) ranges from –100 to 100

**Examples**

POWer:POWer1:SWITCHINGLOSS:VLEVELAbs 2 sets the voltage level (Ton-Start & Stop) value as 2 for the switching loss measurement in the power measurement badge 1.

**POWer:POWer<x>:SWITCHINGLOSS:VLEVELPct**

This command sets or queries the voltage level (Ton-Start & Stop) in percentage for switching loss measurement in the specified power measurement number. The power measurement number is specified by \(<x>\).

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

POWer:POWer<x>:SWITCHINGLOSS:VLEVELPct \(<\text{NR1}>\)

POWer:POWer<x>:SWITCHINGLOSS:VLEVELPct?

**Related Commands**

POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts

**Arguments**

Power\(<x>\) is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

\(<\text{NR1}>\) ranges from 0.0001 to 90

**Examples**

POWer:POWer1:SWITCHINGLOSS:VLEVELPct 7 sets the voltage level (Ton-Start & Stop) value as 7 percentage for the switching loss measurement in the power measurement badge 1.

**POWer:POWer<x>:SWITCHINGLOSS:VSOURce**

This command sets or queries the voltage source for the switching loss measurement in the specified power measurement number. The power measurement number is specified by \(<x>\).

**Conditions**

Requires option PWR or PS2.
Group       Power

Syntax      POWER:POWER<x>:SWITCHINGLOSS:VSOURce {CH<x> | MATH<x> | REF<x>}
            POWER:POWER<x>:SWITCHINGLOSS:VSOURce?

Arguments   Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

            CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

            MATH<x> = A math waveform specifier; <x> is ≥1.

            REF<x> = A reference waveform specifier; <x> is ≥1.

Examples    POWER:POWER1:SWITCHINGLOSS:VSOURce REF1 sets the voltage source for the switching loss measurement in the specified power measurement badge as REF1.

            POWER:POWER2:SWITCHINGLOSS:VSOURce? might return REF3 indicating the voltage source for switching loss measurement of Power 2 power measurement badge.

POWer:POWer<x>:SWITCHINGRIPPLE:INPUTSOurce

This command sets or queries the input source for switching ripple measurement in the specified power measurement number. The power measurement number is specified by <x>.

Conditions Requires option PWR or PS2.

Group       Power

Syntax      POWER:POWER<x>:SWITCHINGRIPPLE:INPUTSOurce {CH<x> | MATH<x> | REF<x>}
            POWER:POWER<x>:SWITCHINGRIPPLE:INPUTSOurce?

Arguments   Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

            CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.
MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

**Examples**

POWer:POWer1:SWITCHINGRIPPLE:INPUTSourcE CH5 sets the input source for switching ripple measurement as CH5 for the power measurement badge Power 1.

**POWer:POWer<x>:SWITCHINGRIPPLE:LFREQuency**

This command sets or queries the switching frequency for switching ripple measurement in the specified power measurement number. The power measurement number is specified by <x>.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

POWer:POWer<x>:SWITCHINGRIPPLE:LFREQuency <NR1>
POWer:POWer<x>:SWITCHINGRIPPLE:LFREQuency?

**Arguments**

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR1> ranges from 50 to 100000

**Examples**

POWer:POWer1:SWITCHINGRIPPLE:SFREQuency 60000 sets the frequency present for line ripple measurement as 60000 Hz.

**POWer:POWer<x>:TURNOFFtime:FREQuency**

This command sets or queries the input frequency used by the AC or DC converter of the specified Turn Off Time measurement.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

POWer:POWer<x>:TURNOFFtime:FREQuency <NR3>
POWer:POWer<x>:TURNOFFtime:FREQuency?
Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the frequency, in Hertz, from 1 Hz to 500 Hz.

Examples

POWer:POWER1:TURNOFFtime:FREQuency 50 sets the frequency value of Turn Off Time measurement 1 to 50 Hz.

POWer:POWER1:TURNOFFtime:FREQuency? might return

POWer:POWER1:TURNOFFtime:FREQuency 350, indicating that the frequency value of Turn Off Time measurement 1 is set to 350 Hz.

POWer:POWER<x>:TURNOFFtime:INPUTLEVel

This command sets or queries the input voltage level of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWER<x>:TURNOFFtime:INPUTLEVel <NR3>

POWer:POWER<x>:TURNOFFtime:INPUTLEVel?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the voltage level, in volts, from –500 V to 500 V.

Examples

POWer:POWER1:TURNOFFtime:INPUTLEVel –350 sets the input voltage level of Turn Off Time measurement 1 to –350 V.

POWer:POWER1:TURNOFFtime:INPUTLEVel? might return

POWer:POWER1:TURNOFFtime:INPUTLEVel 200, indicating that the input voltage level value of Turn Off Time measurement 1 is set to 200 V.

POWer:POWER<x>:TURNOFFtime:INPUTSOurce

This command sets or queries the input source of the specified Turn Off Time measurement.
Commands listed in alphabetical order

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:TURNOFFtime:INPUTSOurce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:INPUTSOurce?
```

**Arguments**

- **Power<x>** is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
- **CH<x>** is the channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.
- **REF<x>** is the Reference waveform specifier ≥ 1. This is the equivalent of the number shown on a Reference waveform badge in the UI.
- **MATH<x>** is the Math waveform specifier ≥ 1. This is the equivalent of the number shown on a Math waveform badge in the UI.

**Examples**

```
POWer:POWer1:TURNOFFtime:INPUTSOurce CH4 sets the input source of Turn Off Time measurement 1 to Channel 4.
POWer:POWer6:TURNOFFtime:INPUTSOurce? might return
POWer:POWer6:TURNOFFtime:INPUTSOurce MATH2, indicating that the input source of Turn Off Time measurement 6 is set to Math waveform 2.
```

**POWer:POWer<x>:TURNOFFtime:MAXTIMe**

This command sets or queries the maximum turn off time of the specified Turn Off Time measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:TURNOFFtime:MAXTIMe <NR3>
POWer:POWer<x>:TURNOFFtime:MAXTIMe?
```

**Arguments**

- **Power<x>** is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
<NR3> is a floating point number that represents the maximum time value, in seconds, in the range 1 second to 500 seconds.

Examples
- `POWER:POWER1:TURNOFFtime:MAXTIME 70` sets the maximum time value of Turn Off Time measurement 1 to 70 seconds.
- `POWER:POWER5:TURNOFFtime:MAXTIME?` might return `POWER:POWER5:TURNOFFtime:MAXTIME 120`, indicating that the maximum time value of Turn Off Time measurement 5 is 120 seconds.

**POWER:POWER<x>:TURNOFFtime:MAXVoltage**

This command sets or queries the maximum voltage of the specified Turn OffTime measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
- `POWER:POWER<x>:TURNOFFtime:MAXVoltage <NR3>`
- `POWER:POWER<x>:TURNOFFtime:MAXVoltage?`

**Arguments**
- `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
- `<NR3>` is a floating point number that represents the maximum voltage in the range 1 V to 500 V.

**Examples**
- `POWER:POWER1:TURNOFFtime:MAXVoltage 50` sets the maximum voltage value of Turn Off Time measurement 1 to 50 volts.
- `POWER:POWER4:TURNOFFtime:MAXVoltage?` might return `POWER:POWER4:TURNOFFtime:MAXVoltage 12`, indicating that the maximum voltage value of Turn Off Time measurement 4 is 12 volts.

**POWER:POWER<x>:TURNOFFtime:NUMOUTputs**

This command sets or queries the number of outputs of the specified Turn Off Time power measurement.

**Conditions**
Requires option PWR or PS2.
Group: Power

Syntax:

POWer:POWer<x>:TURNOFFtime:NUMOUTputs
{ONE|TWO|THREE|FOUR|FIVE|SIX|SEVEN}
POWer:POWer<x>:TURNOFFtime:NUMOUTputs?

Related Commands:

POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURce
POWer:POWer<x>:TURNOFFtime:OUTPUT1VOLTage
POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce
POWer:POWer<x>:TURNOFFtime:OUTPUT2VOLTage
POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce
POWer:POWer<x>:TURNOFFtime:OUTPUT3VOLTage
POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce
POWer:POWer<x>:TURNOFFtime:OUTPUT4VOLTage
POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce
POWer:POWer<x>:TURNOFFtime:OUTPUT5VOLTage
POWer:POWer<x>:TURNOFFtime:OUTPUT6SOURce
POWer:POWer<x>:TURNOFFtime:OUTPUT6VOLTage
POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce
POWer:POWer<x>:TURNOFFtime:OUTPUT7VOLTage

Arguments:

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

ONE through SEVEN sets the number of outputs for the Turn Off Time measurement.

Examples:

POWer:POWer1:TURNOFFtime:NUMOUTputs TWO sets the number of outputs of Turn Off Time measurement 1 to two.

POWer:POWer3:TURNOFFtime:NUMOUTputs? might return POWer:POWer3:TURNOFFtime:NUMOUTputs FOUR, indicating that the number of outputs of Turn Off Time measurement 3 is set to four inputs.
POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURCe

This command sets or queries the output 1 source of the specified Turn Off Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURCe
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURCe?

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer2:TURNOFFtime:OUTPUT1SOURCe CH2 sets the output 1 source of Turn Off Time measurement 2 to Channel 2.

POWer:POWer1:TURNOFFtime:OUTPUT1SOURCe? might return
POWer:POWer1:TURNOFFtime:OUTPUT1SOURCe CH4, indicating that the output 1 source of Turn Off Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT1VOLTage

This command sets or queries the output 1 voltage level of the of the specified Turn Off Time power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

POWer:POWer<x>:TURNOFFtime:OUTPUT1VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT1VOLTage?
Related Commands

POWer:POWer<x>:TURNOFFtime NUMOUTputs
POWer:POWer<x>:TURNOFFtime OUTPUT1SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer7:TURNOFFtime:OUTPUT1VOLTage -2550 sets the output 1 voltage value of Turn Off Time power measurement 7 to –2,550 volts.

POWer:POWer3:TURNOFFtime:OUTPUT1VOLTage? might return
POWer:POWer3:TURNOFFtime:OUTPUT1VOLTage 1000, indicating that the output 1 voltage value of Turn Off Time power measurement 3 is set to 1,000 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce

This command sets or queries the output 2 source of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer3:TURNOFFtime:OUTPUT2SOURce CH2 sets the output 2 source of Turn Off Time measurement 3 to Channel 2.
POWer:POWer5:TURNOFFtime:OUTPUT2SOURCe? might return POWER:POWer5:TURNOFFtime:OUTPUT2SOURCe CH8, indicating that the output 2 source of Turn Off Time measurement 5 is Channel 8.

POWer:POWer<x>:TURNOFFtime:OUTPUT2VOLTage

This command sets or queries the output 2 voltage level of the specified Turn Off Time power measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:TURNOFFtime:OUTPUT2VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT2VOLTage?

**Related Commands**
POWer:POWer<x>:TURNOFFtime:NUMOUTputs
POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURCe

**Arguments**
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

**Examples**
POWER:POWer3:TURNOFFtime:OUTPUT2VOLTage -550 sets the output 2 voltage value of Turn Off Time power measurement 3 to –550 volts.

POWER:POWer1:TURNOFFtime:OUTPUT2VOLTage? might return POWER:POWer1:TURNOFFtime:OUTPUT2VOLTage 100, indicating that the output 2 voltage value of Turn Off Time power measurement 1 is set to 100 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURCe

This command sets or queries the output 3 source of the specified Turn Off Time measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power
Commands listed in alphabetical order

Syntax
POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce?

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples
POWer:POWer3:TURNOFFtime:OUTPUT3SOURce CH8 sets the output 3 source of Turn Off Time measurement 3 to Channel 8.

POWer:POWer1:TURNOFFtime:OUTPUT3SOURce? might return
POWer:POWer1:TURNOFFtime:OUTPUT3SOURce CH4, indicating that the output 3 source of Turn Off Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT3VOLTage

This command sets or queries the output 3 voltage level of the specified Turn Off Time power measurement.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWer:POWer<x>:TURNOFFtime:OUTPUT3VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT3VOLTage?

Related Commands
POWer:POWer<x>:TURNOFFtime NUMOUTputs
POWer:POWer<x>:TURNOFFtime OUTPUT3SOURce

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of −6,000 volts to +6,000 volts.
Examples

POWER:POWER4:TURNOFFtime:OUTPUT3VOLTage -50 sets the output 3 voltage value of Turn Off Time power measurement 4 to –50 volts.

POWER:POWER8:TURNOFFtime:OUTPUT3VOLTage? might return
POWER:POWER8:TURNOFFtime:OUTPUT3VOLTage -200, indicating that the output 3 voltage value of Turn Off Time power measurement 8 is set to –200 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce

This command sets or queries the output 4 source of the specified Turn Off Time measurement.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax

POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWER:POWER3:TURNOFFtime:OUTPUT4SOURce CH2 sets the output 4 source of Turn Off Time measurement 3 is Channel 2.

POWER:POWER10:TURNOFFtime:OUTPUT4SOURce? might return
POWER:POWER10:TURNOFFtime:OUTPUT4SOURce REF4, indicating that the output 4 source of Turn Off Time measurement 10 is Reference waveform 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT4VOLTage

This command sets or queries the output 4 voltage level of the specified Turn Off Time power measurement.
Commands listed in alphabetical order

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:TURNOFFtime:OUTPUT4VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT4VOLTage?
```

**Related Commands**

```
POWer:POWer<x>:TURNOFFtime NUMOUTputs
POWer:POWer<x>:TURNOFFtime OUTPUT4SOURce
```

**Arguments**

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR2>` sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

**Examples**

```
POWer:POWer8:TURNOFFtime:OUTPUT4VOLTage 50 sets the output 4 voltage value of Turn Off Time power measurement 8 to 50 volts.
POWer:POWer4:TURNOFFtime:OUTPUT4VOLTage? might return POWer:POWer4:TURNOFFtime:OUTPUT4VOLTage 1000, indicating that the output 4 voltage value of Turn Off Time power measurement 4 is set to 1,000 volts.
```

**POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce**

This command sets or queries the output 5 source of the specified Turn Off Time measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce?
```

**Arguments**

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.
Commands listed in alphabetical order

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer3:TURNOFFtime:OUTPUT5SOURCE CH2 sets the output 5 source of Turn Off Time measurement 3 to Channel 2.

POWer:POWer11:TURNOFFtime:OUTPUT5SOURCE? might return
POWer:POWer11:TURNOFFtime:OUTPUT5SOURCE CH4, indicating that the output 5 source of Turn Off Time measurement 11 is Channel 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT5VOLTage

This command sets or queries the output 5 voltage level of the specified Turn Off Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:TURNOFFtime:OUTPUT5VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT5VOLTage?

Related Commands

POWer:POWer<x>:TURNOFFtime:NUMOUTputs
POWer:POWer<x>:TURNOFFtime OUTPUT5SOURCE

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer2:TURNOFFtime:OUTPUT5VOLTage -255 sets the output 5 voltage value of Turn Off Time power measurement 2 to –250 volts.

POWer:POWer3:TURNOFFtime:OUTPUT5VOLTage? might return
POWer:POWer3:TURNOFFtime:OUTPUT5VOLTage 100, indicating that the output 5 voltage value of Turn Off Time power measurement 3 is set to 100 volts.
POWER:POWer<x>:TURNOFFtime:OUTPUT6SOURce

This command sets or queries the output 6 source of the specified Turn Off Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax POWER:POWer<x>:TURNOFFtime:OUTPUT6SOURce
{CH<x>|REF<x>|MATH<x>}
POWER:POWer<x>:TURNOFFtime:OUTPUT6SOURce?

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples POWER:POWer3:TURNOFFtime:OUTPUT6SOURce CH2 sets the output 6 source of Turn Off Time measurement 3 to Channel 2.
POWER:POWer1:TURNOFFtime:OUTPUT6SOURce? might return POWER:POWer1:TURNOFFtime:OUTPUT6SOURce CH4, indicating that the output 6 source of Turn Off Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT6VOLTage

This command sets or queries the output 6 voltage level of the specified Turn Off Time power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax POWER:POWer<x>:TURNOFFtime:OUTPUT6VOLTage <NR2>
POWER:POWer<x>:TURNOFFtime:OUTPUT6VOLTage?
Related Commands

POWer:POWer<x>:TURNOFFtime:NUMOUTputs
POWer:POWer<x>:TURNOFFtime:OUTPUT6SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer5:TURNOFFtime:OUTPUT6VOLTage 1550 sets the output 6 voltage value of Turn Off Time power measurement 5 to 1,550 volts.

POWer:POWer2:TURNOFFtime:OUTPUT6VOLTage? might return
POWer:POWer2:TURNOFFtime:OUTPUT6VOLTage –100, indicating that the output 6 voltage value of Turn Off Time power measurement 2 is set to –100 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce

This command sets or queries the output 7 source of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer3:TURNOFFtime:OUTPUT7SOURce CH2 sets the output 7 source of Turn Off Time measurement 3 is Channel 2.
POWer:POWer1:TURNOFFtime:OUTPUT7SOURCE? might return POWER:POWer1:TURNOFFtime:OUTPUT7SOURCE CH4, indicating that the output 7 source of Turn Off Time measurement 1 is channel 4.

**POWer:POWer<x>:TURNOFFtime:OUTPUT7VOLTage**

This command sets or queries the output 7 voltage level of the specified Turn Off Time power measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWER:POWer<x>:TURNOFFtime:OUTPUT7VOLTage <NR2>
POWER:POWer<x>:TURNOFFtime:OUTPUT7VOLTage?

**Related Commands**
POWER:POWer<x>:TURNOFFtime NUMOUTPUTs
POWER:POWer<x>:TURNOFFtime OUTPUT7SOURCE

**Arguments**
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

**Examples**
POWER:POWer2:TURNOFFtime:OUTPUT7VOLTage -2100 sets the output 7 voltage value of Turn Off Time power measurement 2 to –2,100 volts.

POWER:POWer3:TURNOFFtime:OUTPUT7VOLTage? might return POWER:POWer3:TURNOFFtime:OUTPUT7VOLTage 450, indicating that the output 7 voltage value of Turn Off Time power measurement 3 is set to 450 volts.

**POWer:POWer<x>:TURNOFFtime:TYPE**

This command sets or queries the type of AC/DC converter used in the specified Turn Off Time power measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power
Commands listed in alphabetical order

Syntax

POWer:POWer<x>:TURNOFFtime:TYPE {DCDC|ACDC}
POWer:POWer<x>:TURNOFFtime:TYPE?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

DCDC sets the measurement to use a DC to DC converter.

ACDC sets the measurement to use an AC to DC converter.

Examples

POWer:POWer1:TURNOFFtime:TYPE ACDC sets Turn Off time measurement 1 to use an AC to DC converter.

POWer:POWer5:TURNOFFtime:TYPE? might return
POWer:POWer5:TURNOFFtime:TYPE DCDC, indicating that the converter type for Turn Off Time measurement 5 is set to use a DC to DC converter.

POWer:POWer<x>:TURNONtime:FREQuency

This command sets or queries the input frequency used by the AC or DC converter of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:TURNONtime:FREQuency <NR3>
POWer:POWer<x>:TURNONtime:FREQuency?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the frequency, in Hertz, from 1 Hz to 500 Hz.

Examples

POWer:POWer1:TURNONtime:FREQuency 50 sets the frequency value of the AC/DC converter in Turn On Time measurement 1 to 50 Hz.

POWer:POWer4:TURNONtime:FREQuency? might return
POWer:POWer4:TURNONtime:FREQuency 200, indicating that the frequency value of the AC/DC converter in Turn On Time measurement 4 is set to 200 Hz.
POWer:POWer<x>:TURNONtime:INPUTLEVEL

This command sets or returns the input voltage level of the specified Turn On Time measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
```
POWer:POWer<x>:TURNONtime:INPUTLEVEL <NR3>
POWer:POWer<x>:TURNONtime:INPUTLEVEL?
```

**Arguments**
- **Power<x>** is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
- **<NR3>** is a floating point number that represents the voltage level, in volts, from \(-500\) \(V\) to \(500\) \(V\).

**Examples**
```
POWer:POWer12:TURNONtime:INPUTLEVEL \(-350\) sets the input voltage level of Turn On Time measurement 12 to \(-350\) \(V\).
POWer:POWer4:TURNONtime:INPUTLEVEL? might return
POWer:POWer4:TURNONtime:INPUTLEVEL \(230\), indicating that the input voltage level value of Turn ON Time measurement 4 is set to \(230\) \(V\).
```

POWer:POWer<x>:TURNONtime:INPUTSOURCE

This command sets or queries the input source of the specified Turn On Time measurement.

**Group**
Power

**Syntax**
```
POWer:POWer<x>:TURNONtime:INPUTSOURCE
POWer:POWer<x>:TURNONtime:INPUTSOURCE?
```

**Arguments**
- **Power<x>** is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
- **CH<x>** is the channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.
- **REF<x>** is the Reference waveform specifier \(\geq 1\). This is the equivalent of the number shown on a Reference waveform badge in the UI.
MATH<x> is the Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

**Examples**

```
POWer:POWer1:TURNONtime:INPUTSource CH2 sets the input source of Turn On Time measurement 1 to Channel 2.
```

```
POWer:POWer2:TURNONtime:INPUTSource? might return
POWer:POWer2:TURNONtime:INPUTSource MATH6, indicating that the input source of Turn On Time measurement 2 is Math waveform 6.
```

**POWer:POWer<x>:TURNONtime:MAXTIMe**

This command sets or returns the maximum turn on time of the specified Turn On Time measurement.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

```
POWer:POWer<x>:TURNONtime:MAXTIMe <NR3>
POWer:POWer<x>:TURNONtime:MAXTIMe?
```

**Arguments**

- `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
- `<NR3>` is a floating point number that represents the maximum time value, in seconds, in the range 1 second to 500 seconds.

**Examples**

```
POWer:POWer3:TURNONtime:MAXTIMe 30 sets the maximum time value of Turn On Time measurement 3 to 30 seconds.
```

```
POWer:POWer5:TURNONtime:MAXTIMe? might return
POWer:POWer5:TURNONtime:MAXTIMe 120, indicating that the maximum time value of Turn On Time measurement 5 is set to 120 seconds.
```

**POWer:POWer<x>:TURNONTIME:MAXVoltag**e

This command sets or returns the maximum voltage setting of the specified Turn On Time measurement.

**Conditions**

Requires option PWR or PS2.
POWer:POWer<x>:TURNONtime:MAXVoltage <NR3>
POWer:POWer<x>:TURNONtime:MAXVoltage?

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the maximum voltage in the range 1 V to 500 V.

Examples
POWer:POWer1:TURNONtime:MAXVoltage 50 sets the maximum voltage value of Turn On Time measurement 1 to 50 volts.

POWer:POWer4:TURNONtime:MAXVoltage? might return
POWer:POWer4:TURNONtime:MAXVoltage 12, indicating that the maximum voltage value of Turn On Time measurement 4 is set to 12 volts.

POWer:POWer<x>:TURNONtime:NUMOUTputs

This command sets or queries the number of outputs for the specified Turn On Time power measurement.

Conditions
Requires option PWR or PS2.

Related Commands
POWer:POWer<x>:TURNONtime:INPUTSOurce
POWer:POWer<x>:TURNONtime:OUTPUT1SOURce
POWer:POWer<x>:TURNONtime:OUTPUT1VOLTage
POWer:POWer<x>:TURNONtime:OUTPUT2SOURce
POWer:POWer<x>:TURNONtime:OUTPUT2VOLTage
POWer:POWer<x>:TURNONtime:OUTPUT3SOURce
POWer:POWer<x>:TURNONtime:OUTPUT3VOLTage
POWeR:POWeR<x>:TURNONtime:OUTPUT4SOURce
POWeR:POWeR<x>:TURNONtime:OUTPUT4VOLTage
POWeR:POWeR<x>:TURNONtime:OUTPUT5SOURce
POWeR:POWeR<x>:TURNONtime:OUTPUT5VOLTage
POWeR:POWeR<x>:TURNONtime:OUTPUT6SOURce
POWeR:POWeR<x>:TURNONtime:OUTPUT6VOLTage
POWeR:POWeR<x>:TURNONtime:OUTPUT7SOURce
POWeR:POWeR<x>:TURNONtime:OUTPUT7VOLTage

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

ONE through SEVEN sets the number of outputs for the specified Turn On Time power measurement.

Examples
POWeR:POWeR1:TURNONtime:NUMOUTputs TWO sets the number of outputs of Turn On Time measurement 1 to two.
POWeR:POWeR3:TURNONtime:NUMOUTputs? might return POWeR:POWeR3:TURNONtime:NUMOUTputs FOUR, indicating that the number of outputs of Turn On Time measurement 3 is set to four inputs.

POWeR:POWeR<x>:TURNONtime:OUTPUT1SOURce

This command sets or queries the output 1 source of the specified Turn On Time measurement.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax
POWeR:POWeR<x>:TURNONtime:OUTPUT1SOURce
{CH<x>|REF<x>|MATH<x>}
POWeR:POWeR<x>:TURNONtime:OUTPUT1SOURce?

Arguments
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.
Commands listed in alphabetical order

**REF<x>** A Reference waveform specifier $\geq 1$. This is the equivalent of the number shown on a Reference waveform badge in the UI.

**MATH<x>** A Math waveform specifier $\geq 1$. This is the equivalent of the number shown on a Math waveform badge in the UI.

**Examples**

```
POWER:POWER5:TURNONtime:OUTPUT1SOURCE CH2 sets the output 1 source of Turn On Time measurement 5 to Channel 2.
POWER:POWER1:TURNONtime:OUTPUT1SOURCE? might return
POWER:POWER1:TURNONtime:OUTPUT1SOURCE CH4, indicating that the output 1 source of Turn On Time measurement 1 is Channel 4.
```

**POWer:POWer<x>:TURNONtime:OUTPUT1VOLTage**

This command sets or queries the output 1 voltage level of the specified Turn On Time power measurement.

**Conditions** Requires option PWR or PS2.

**Group** Power

**Syntax**

```
POWER:POWER<x>:TURNONtime:OUTPUT1VOLTage <NR2>
POWER:POWER<x>:TURNONtime:OUTPUT1VOLTage?
```

**Related Commands**

```
POWER:POWER<x>:TURNONtime:NUMOUTputs
POWER:POWER<x>:TURNONtime:OUTPUT1SOURCE
```

**Arguments**

*Power<x>* is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of $-6,000$ volts to $+6,000$ volts.

**Examples**

```
POWER:POWER1:TURNONtime:OUTPUT1VOLTage -2550 sets the output 1 voltage value of Turn On Time power measurement 1 to $-2,550$ volts.
POWER:POWER3:TURNONtime:OUTPUT1VOLTage? might return
POWER:POWER3:TURNONtime:OUTPUT1VOLTage 1000, indicating that the output 1 voltage value of Turn On Time power measurement 3 is set to $1,000$ volts.
```
POWer:POWer<x>:TURNONtime:OUTPUT2SOURce

This command sets or queries the output 2 source of the specified Turn On Time measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:TURNONtime:OUTPUT2SOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT2SOURce?

**Arguments**
Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

**Examples**
POWer:POWer2:TURNONtime:OUTPUT2SOURce CH2 sets the output 2 source of Turn On Time measurement 2 to Channel 2.
POWer:POWer1:TURNONtime:OUTPUT2SOURce? might return POWer:POWer1:TURNONtime:OUTPUT2SOURce CH4, indicating that the output 2 source of Turn On Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNONtime:OUTPUT2VOLTage

This command sets or queries the output 2 voltage level of the specified Turn On Time power measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
POWer:POWer<x>:TURNONtime:OUTPUT2VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT2VOLTage?
Related Commands

POWer:POWer<x>:TURNONtime:NUMOUTputs
POWer:POWer<x>:TURNONtime:OUTPUT2SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer7:TURNONtime:OUTPUT2VOLTage -2550 sets the output 2 voltage value of Turn On Time power measurement 7 to –2,550 volts.

POWer:POWer1:TURNONtime:OUTPUT2VOLTage? might return POWer:POWer1:TURNONtime:OUTPUT2VOLTage 120, indicating that the output 2 voltage value of Turn On Time power measurement 1 is set to 120 volts.

POWer:POWer<x>:TURNONtime:OUTPUT3SOURce

This command sets or queries the output 3 source of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:TURNONtime:OUTPUT3SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT3SOURce?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer3:TURNONtime:OUTPUT3SOURce CH2 sets the output 3 source of Turn On Time measurement 3 to Channel 2.

POWer:POWer<x>:TURNOntime:OUTPUT3VOLTage

This command sets or queries the output 3 voltage level of the specified Turn On Time power measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**
```
POWer:POWer<x>:TURNOntime:OUTPUT3VOLTage <NR2>
POWer:POWer<x>:TURNOntime:OUTPUT3VOLTage?
```

**Related Commands**
POWer:POWer<x>:TURNOntime NUMOUTputs
POWer:POWer<x>:TURNOntime OUTPUT3SOURCE

**Arguments**
- `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.
- `<NR2>` sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

**Examples**
```
POWer:POWer7:TURNOntime:OUTPUT3VOLTage -550 sets the output 3 voltage value of Turn On Time power measurement 7 to –550 volts.
```
```
POWer:POWer3:TURNOntime:OUTPUT3VOLTage? might return POWER:POWer3:TURNOntime:OUTPUT3VOLTage –200, indicating that the output 3 voltage value of Turn On Time power measurement 3 is set to –200 volts.
```

POWer:POWer<x>:TURNOntime:OUTPUT4SOURCE

This command sets or queries the output 4 source of the specified Turn On Time measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power
Commands listed in alphabetical order

Syntax

POWer:POWer<x>:TURNONtime:OUTPUT4SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT4SOURce?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer3:TURNONtime:OUTPUT4SOURce CH2 sets the output 4 source of Turn On Time measurement 3 to Channel 2.

POWer:POWer1:TURNONtime:OUTPUT4SOURce? might return
POWer:POWer1:TURNONtime:OUTPUT4SOURce MATH3, indicating that the output 4 source of Turn On Time measurement 1 is Math waveform 3.

POWer:POWer<x>:TURNONtime:OUTPUT4VOLTage

This command sets or queries the output 4 voltage level of the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:TURNONtime:OUTPUT4VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT4VOLTage?

Related Commands

POWer:POWer<x>:TURNONtime:NUMOUTputs
POWer:POWer<x>:TURNONtime:OUTPUT4SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of −6,000 volts to +6,000 volts.
Commands listed in alphabetical order

Examples

POWER:POWER8:TURNONtime:OUTPUT4VOLTage 50 sets the output 4 voltage value of Turn On Time power measurement 8 to 50 volts.

POWER:POWER2:TURNONtime:OUTPUT4VOLTage? might return
POWER:POWER2:TURNONtime:OUTPUT4VOLTage 1000, indicating that the output 4 voltage value of Turn On Time power measurement 2 is set to 1,000 volts.

POWer:POWer<x>:TURNONtime:OUTPUT5SOURce

This command sets or queries the output 5 source of the specified Turn On Time measurement.

Conditions
Requires option PWR or PS2.

Group
Power

Syntax

POWER:POWER<x>:TURNONtime:OUTPUT5SOURce
{CH<x>|REF<x>|MATH<x>}
POWER:POWER<x>:TURNONtime:OUTPUT5SOURce?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWER:POWER3:TURNONtime:OUTPUT5SOURce CH2 sets the output 5 source of Turn On Time measurement 3 to Channel 2.

POWER:POWER1:TURNONTIME:OUTPUT5SOURce? might return
POWER:POWER1:TURNONtime:OUTPUT5SOURce CH4, indicating that the output 5 source of Turn On Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNONtime:OUTPUT5VOLTage

This command sets or queries the output 5 voltage level of the specified Turn On Time power measurement.
Commands listed in alphabetical order

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:TURNONtime:OUTPUT5VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT5VOLTage?
```

**Related Commands**

POWer:POWer<x>:TURNONtime:NUMOUTputs

POWer:POWer<x>:TURNONtime:OUTPUT5SOURce

**Arguments**

*Power<x>* is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

**Examples**

POWer:POWer2:TURNONtime:OUTPUT5VOLTage -255 sets the output 5 voltage value of Turn On Time power measurement 2 to –250 volts.

POWer:POWer3:TURNONtime:OUTPUT5VOLTage? might return POWER:POWer3:TURNONtime:OUTPUT5VOLTage 100, indicating that the output 5 voltage value of Turn On Time power measurement 3 is set to 100 volts.

**POWer:POWer<x>:TURNONtime:OUTPUT6SOURce**

This command sets or queries the output 6 source of the specified Turn On Time measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

```
POWer:POWer<x>:TURNONtime:OUTPUT6SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT6SOURce?
```

**Arguments**

*Power<x>* is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<CH<x>> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.
REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

**Examples**

POWer:POWer3:TURNOnTime:OUTPUT6SOURce CH2 sets the output 6 source of Turn On Time measurement 3 to Channel 2.

POWer:POWer8:TURNOnTime:OUTPUT6SOURce? might return

POWer:POWer8:TURNOnTime:OUTPUT6SOURce CH4, indicating that the output 6 source of Turn On Time measurement 8 is set to Channel 4.

**POWer:POWer<x>:TURNOnTime:OUTPUT6VOLTage**

This command sets or queries the output 6 voltage level of the specified Turn On Time power measurement.

**Conditions**

Requires option PWR or PS2.

**Group**

Power

**Syntax**

POWer:POWer<x>:TURNOnTime:OUTPUT6VOLTage <NR2>

POWer:POWer<x>:TURNOnTime:OUTPUT6VOLTage?

**Related Commands**

POWer:POWer<x>:TURNOnTime NUMOUTputs

POWer:POWer<x>:TURNOnTime OUTPUT6SOURce

**Arguments**

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

**Examples**

POWer:POWer5:TURNOnTime:OUTPUT6VOLTage 1550 sets the output 6 voltage value of Turn On Time power measurement 5 to 1,550 volts.

POWer:POWer2:TURNOnTime:OUTPUT6VOLTage? might return

POWer:POWer2:TURNOnTime:OUTPUT6VOLTage –100, indicating that the output 6 voltage value of Turn On Time power measurement 2 is set to –100 volts.
**POWer:POWer<x>:TURNONtime:OUTPUT7SOURce**

This command sets or queries the output 7 source of the specified Turn On Time measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

POWer:POWer<x>:TURNONtime:OUTPUT7SOURce
{CH<x>|REF<x>|MATH<x>}

POWer:POWer<x>:TURNONtime:OUTPUT7SOURce?

**Arguments**

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

**Examples**

POWer:POWer3:TURNONtime:OUTPUT6SOURce CH1 sets the output 7 source of Turn On Time measurement 3 to Channel 1.

POWer:POWer7:TURNONtime:OUTPUT6SOURce? might return POWer:POWer7:TURNONtime:OUTPUT6SOURce REF4, indicating that the output 7 source of Turn On Time measurement 7 is set to Reference waveform 4.

**POWer:POWer<x>:TURNONtime:OUTPUT7VOLTage**

This command sets or queries the output 7 voltage level of the specified Turn On Time power measurement.

**Conditions**
Requires option PWR or PS2.

**Group**
Power

**Syntax**

POWer:POWer<x>:TURNONtime:OUTPUT7VOLTage <NR2>

POWer:POWer<x>:TURNONtime:OUTPUT7VOLTage?
Related Commands

POWer:POWer<x>:TURNONtime:NUMOUTputs
POWer:POWer<x>:TURNONtime:OUTPUT7SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer12:TURNONtime:OUTPUT7VOLTage -2100 sets the output 7 voltage value of Turn On Time power measurement 12 to –2,100 volts.

POWer:POWer3:TURNONtime:OUTPUT7VOLTage? might return
POWer:POWer3:TURNONtime:OUTPUT7VOLTage 450, indicating that the output 7 voltage value of Turn On Time power measurement 3 is set to 450 volts.

POWer:POWer<x>:TURNONtime:TYPE

This command sets or queries the type of AC/DC converter used in the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

POWer:POWer<x>:TURNONtime:TYPE {DCDC|ACDC}
POWer:POWer<x>:TURNONtime:TYPE?

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

DCDC sets the measurement to use a DC to DC converter.

ACDC sets the measurement to use an AC to DC converter.

Examples

POWer:POWer3:TURNONtime:TYPE ACDC sets Turn On time measurement 3 to use an AC to DC converter.

POWer:POWer12:TURNONtime:TYPE? might return
POWer:POWer12:TURNONtime:TYPE DCDC, indicating that the converter type for Turn On Time measurement 12 is set to use a DC to DC converter.
POWer:POWer<x>:TYPe

This command sets or queries the measurement type of the specified power measurement number. If the measurement number does not exist, this command creates a new power measurement, assigns the specified measurement number to the new measurement, and then assigns the measurement type to the new measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax POWer:POWer<x>:TYPe <Measurement Type>
POWer:POWer<x>:TYPe?

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<Measurement Type> = CYCLEAmp | CYCLEBase | CYCLEMAx | CYCLEMin | CYCLEPKpk | CYCLETop | DIDT | DVDT | EFFICIENCY | FREQuency | HARMonics | INDUCTANCE | INPUTCAP | INRUSHcurrent | IVSINTEGRALV | LINERIpple | MAGNETICLOSS | MAGPROPERTY | NDUTYCycle | NPULSEWidth | PDUTYCycle | PERiod | POWERQUALity | PPULSEWidth | RDSON | SOA | SWITCHINGLOss | SWITCHINGRipple | TURNOFFtime | TURNOOnTime | CLRESPONSE | PSRR

Examples POWER:POWER6:TYPe INDUCTANCE sets power measurement 6 to be an Inductance measurement.

POWER:POWER1:TYPe? might return INPUTCAP, indicating that power measurement 1 is an Input Capacitance measurement.

POWERTABle:ADDNew (No Query Form)

This command adds the power harmonics table. The power measurement badge is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax POWERTABle:ADDNew TABLE<x>
Arguments <x> must be greater than or equal to one.

Examples POWERTABLE:ADDnew "TABLE1" adds power harmonics table.

**POWERTABLE:DELETE (No Query Form)**

This command deletes the power harmonics table.

Conditions Requires option PWR or PS2.

Group Power

Syntax POWERTABLE:DELETE TABLE<x>

Arguments <x> must be greater than or equal to one.

Examples POWERTABLE:DELETE "TABLE1" deletes TABLE1 power harmonics table.

**POWERTABLE:LIST? (Query Only)**

This command lists all defined power harmonics table.

Conditions Requires option PWR or PS2.

Group Power

Syntax POWERTABLE:LIST?

Examples POWERTABLE:LIST? might return TABLE1 indicating the defined power harmonics table.

**PSC**

This command sets and queries the power-on status flag that controls the automatic power-on handling of the DESER, SRER, and ESER registers. When *PSC is true, the DESER register is set to 255 and the SRER and ESER registers...
are set to 0 at power-on. When *PSC is false, the current values in the DESER, SRER, and ESER registers are preserved in nonvolatile memory when power is shut off and are restored at power-on.

**Group** Status and Error

**Syntax**
*
PSC {<NR1>|OFF|ON}
*PSC?

**Related Commands**
DESE
*ESE
FACTory
*RST
*SRE

**Arguments**

<NR1> = 0 sets the power-on status clear flag to false, disables the power-on clear and allows the instrument to possibly assert SRQ after power-on; any other value sets the power-on status clear flag to true, enabling the power-on status clear and prevents any SRQ assertion after power on.

OFF sets the power-on status clear flag to false, disables the power-on clear and allows the instrument to possibly assert SRQ after power-on.

ON sets the power-on status clear flag to true, enabling the power-on status clear and prevents any SRQ assertion after power on.

**Examples**

*PSC 0 sets the power-on status clear flag to false.

*PSC? might return 1 to indicate that the power-on status clear flag is set to true.

**PUD**

This command sets or queries a string of Protected User Data. This data is protected by the PASSWord command. You can modify it only by first entering the correct password. This password is not necessary to query the data.

**Group** Status and Error

**Syntax**
*
PUD {<Block>|<QString>}
*PUD?

2-754 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
Commands listed in alphabetical order

**Related Commands**

PASSWord

**Arguments**

<Block> is a block containing up to 100 characters.

<QString> is a string containing up to 100 characters.

**Examples**

*PUD #229This instrument belongs to me stores the string "This instrument belongs to me" in the user protected data area.

*PUD? might return #221PROPERTY OF COMPANY X.

**RECALL:SESSION (No Query Form)**

Restores the state of the instrument, including reference waveforms, from a saved session file.

**Group**

Save and Recall

**Syntax**

RECALL:SESSION <file_path>

**Arguments**

=file_path= is the file path that specifies the location of the specified instrument session file.

If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a drive designator (such as C), then the file name is interpreted as a full path.

- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

**Returns**

Recalls the instrument session from the specified session file.

**Examples**

RECALL:SESSION "TEK00000.TSS" recalls the setup from the file TEK00000.TSS in the current working directory.
RECALL:SETUp (No Query Form)
This command (no query form) returns stored or factory settings to the instrument from a copy of the settings stored in memory. This command performs the same function as selecting Recall from the File menu, and then choosing the Setup button.

Group: Save and Recall

Syntax: RECALL:SETUp {FACTory|<file_path>}

Related Commands: FACTory, *RST

Arguments:
- FACTory: restores the factory setup. Performs the same operation as the FACTory command.
- <file_path>: specifies a location for an instrument setup file. <file_path> is a quoted string that defines the file name and path. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:
  - Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
  - Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples:
- RECALL:SETUP FACTORY recalls (and makes current) the instrument setup to its factory defaults.
- RECALL:SETUP "TEK00000.SET" recalls the setup from the file TEK00000.SET in the default directory for setups.

RECALL:WAVEform (No Query Form)
This command recalls a stored waveform to a reference memory location.

Group: Save and Recall
Commands listed in alphabetical order

Syntax  RECALL:WAVEForm <source file>,<destination>

Arguments  <source file> is the source file. The file is expected to be located in a directory relative to the current working directory (specified by FILESystem CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

<destination> is REF<x> which specifies a reference to create from the recalled waveform data file.

Examples  RECALL:WAVEForm "TEK00000.ISF",REF1 recalls the waveform stored in the file named TEK00000.ISF from the current directory to reference 1.

REF:ADDNew (No Query Form)

This command adds the specified reference. Argument is of the form "REF<NR1> ", where NR1 ≥ 1.

Group  Vertical

Syntax  REF:ADDNew <QString>

Arguments  <QString> is the specified reference. Argument is of the form "REF<NR1> ", where NR1 ≥ 1.

Examples  REF:ADDNEW "REF2" adds reference 2 to the display.

REF:DELete (No Query Form)

Deletes the specified reference. Argument is of the form "REF<NR1> ", where NR1 ≥ 1.

Conditions  Vertical
Commands listed in alphabetical order

<table>
<thead>
<tr>
<th>Group</th>
<th>Vertical</th>
</tr>
</thead>
</table>

**Syntax**

REF:DELETE <QString>

**Arguments**

<QString> is the specified reference. Argument is of the form "REF<NR1>", where NR1 ≥ 1.

**Examples**

REF:DELETE "REF2" deletes reference 2 from the display.

**REF:LIST? (Query Only)**

This command returns a comma separated list of all currently defined references.

<table>
<thead>
<tr>
<th>Group</th>
<th>Vertical</th>
</tr>
</thead>
</table>

**Syntax**

REF:LIST?

**Returns**

All currently defined references.

**Examples**

REF:LIST? might return :REF:LIST REF1,REF2 indicating references 1 and 2 are defined.

**REF:REF<x>:DESKew**

This command sets or queries the deskew value used for the specified reference.

<table>
<thead>
<tr>
<th>Group</th>
<th>Vertical</th>
</tr>
</thead>
</table>

**Syntax**

REF:REF<x>:DESKew <NR3>

**Arguments**

<NR3> is the deskew value used for the specified reference.

**Examples**

REF:REF1:DESKew -1.5e-9 sets the deskew value to -1.5 ns.

REF:REF1:DESKew? might return :REF:REF1:DESKew 1.5200E-9 indicating the deskew value is 1.52 ns.
REF:REF<x>:LABel:COLor

This command sets or queries the color of the specified ref label.

Group Vertical

Syntax REF:REF<x>:LABel:COLOR <QString>

Arguments <QString> is the label. To return the color to the default color, send an empty string as in this example: :REF:REF1:LABEL:COLOR ""

Examples REF:REF1:LABel:COLOR "#FFFF00" sets the font color to yellow.
REF:REF1:LABEL:COLOR? might return :REF:REF1:LABEL:COLOR "#FF0000" indicating the font color is red.

REF:REF<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified reference label.

Group Vertical

Syntax REF:REF<x>:LABel:FONT:BOLD {<NR1>|OFF|ON}

Arguments <NR1> = 0 disables bold font; any other value turns this feature on.
OFF disables bold font.
ON enables bold font.

Examples REF:REF1:LABel:FONT:BOLD ON turns on the bold font.

REF:REF<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the specified reference label.

Group Vertical
Commands listed in alphabetical order

Syntax

REF:REF<x>:LABel:FONT:ITALiC {<NR1>|OFF|ON}

Arguments

<NR1> = 0 disables italic font, any other value turns this feature on.
OFF disables italic font.
ON enables italic font.

Examples

REF:REF1:LABel:FONT:ITALiC ON turns on the italic font.
REF:REF1:LABel:FONT:ITALiC? might return :REF:REF1:LABEL:FONT:ITALIC 0 indicating the italic font is off.

REF:REF<x>:LABel:FONT:SIZE

This command sets or queries the font size of the specified reference label.

Group

Vertical

Syntax

REF:REF<x>:LABel:FONT:SIZE <NR1>

Arguments

<NR1> is the font size of the label.

Examples

REF:REF1:LABel:FONT:SIZE 20 sets the font size to 20 points.

REF:REF<x>:LABel:FONT:TYPE

This command sets or queries the font type of the specified reference label, such as Arial or Times New Roman.

Group

Vertical

Syntax

REF:REF<x>:LABel:FONT:TYPE <QString>

Arguments

<QString> is the font type.
Commands listed in alphabetical order

**Examples**

```
REF:REF1:LABEL:FONT:TYPE "Monospace" specifies a mono spaced font.
```

**REF:REF<x>:LABEL:FONT:UNDERline**

This command sets or queries the underline state of the specified reference label.

**Group**
Vertical

**Syntax**
```
REF:REF<x>:LABEL:FONT:UNDERline {<NR1>|OFF|ON}
```

**Arguments**

- `<NR1>` = 0 disables underline font, any other value turns this feature on.
- OFF disables underline font.
- ON enables underline font.

**Examples**

```
REF:REF1:LABEL:FONT:UNDERline ON turns on the underline font.
```

**REF:REF<x>:LABEL:NAMe**

This command sets or queries the label of the specified reference. The reference waveform is specified by `x`.

**Group**
Vertical

**Syntax**
```
REF:REF<x>:LABEL:NAMe <QString>
REF:REF<x>:LABEL:NAMe?
```

**Arguments**

- `<QString>` is the character string that will be used for the reference waveform label name.

**Examples**

```
REF:REF4:LABEL:NAMe "My Reference" sets the label name of Reference 4 waveform to "My Reference".
```

**REF:REF<x>:LABel:XPOS**

This command sets or queries the X-position at which the label (attached to the displayed waveform of the specified reference) is displayed, relative to the left edge of the waveform. The reference waveform is specified by x.

**Group** Vertical

**Syntax**

REF:REF<x>:LABEL:XPOS <NR1>
REF:REF<x>:LABEL:XPOS?

**Arguments**

<NR1> is the location (control in divisions) where the waveform label for the selected reference is displayed, relative to the left edge of the screen.

**Examples**

REF:REF4:LABEL:XPOS 10 moves the waveform label for the Reference 3 waveform, so that it begins 10 divisions to the right of the left edge of the screen.

REF:REF2:LABEL:XPOS? might return :REF:REF2:LABEL:XPOS 1.5, indicating that the x-axis for the Reference 2 waveform is currently 1.5 divisions to the right of the left edge of the screen.

**REF:REF<x>:LABel:YPOS**

This command sets or queries the Y-position of the label (attached to the displayed waveform of the specified reference), relative to the baseline of the waveform. The reference waveform is specified by x.

**Group** Vertical

**Syntax**

REF:REF<x>:LABEL:YPOS <NR1>
REF:REF<x>:LABEL:YPOS?

**Arguments**

<NR1> is the location where the waveform label for the selected reference is displayed, relative to the baseline of the waveform.
Examples  

REF:REF3:LABEL:YPOS -10 moves the waveform label for the Reference 3 waveform 10 vertical units below the baseline of the waveform.

REF:REF2:LABEL:YPOS? might return :REF:REF2:LABEL:YPOS 0, indicating that the waveform label for the Reference 2 waveform is currently located at the baseline of the waveform.

REF:REF<x>:SOUrce

This command sets or queries the filename used by the given reference.

Group  Vertical

Syntax  

REF:REF<x>:SOUrce <QString>

Arguments  

<QString> is the reference file name.

Examples

REF:REF1:SOUrce
"/home/guest/.local/share/Tektronix/TekScope/ FirstRecalledSession/161012_132000_000.wfm" sets the source of the reference.

REF:REF1:SOUrce? might return :REF:REF1:SOUrce
"/home/guest/.local/share/Tektronix/TekScope/ LastRecalledSession/161012_132039_000.wfm".

REF<x>_DALL:LABel:COlor

This command sets or queries the color of the specified digital group. The reference is specified by x.

Group  Digital

Syntax  

REF<x>_DALL:LABEL:COLOR <QString>

Arguments  

<QString> is the color of the digital group label. To return the color to the default color, send an empty string as in this example: :REF5_DALL:LABEL:COLOR "".
Examples  
REF1_DALL:LABEL:COLOR "#FF0000" sets the font color to red.
REF1_DALL:LABEL:COLOR? might return :REF1_DALL:LABEL:COLOR "#FFFF00" indicating the font color is yellow.

**REF<x>_DALL:LABEL:FONT:BOLD**

This command sets or queries the bold state of the specified digital group. The reference is specified by x.

**Group**  
Digital

**Syntax**  
REF<x>_DALL:LABEL:FONT:BOLD {ON|OFF|<NR1>}

**Arguments**  
OFF argument turns off bold font.
ON argument turns on bold font.
<NR1> = 0 turns off bold font; any other value turns on bold font.

**Examples**  
REF1_DALL:LABEL:FONT:BOLD ON sets the font to bold.
REF1_DALL:LABEL:FONT:BOLD? might return :REF1_DALL:LABEL:FONT:BOLD 0 indicating the font is not bold.

**REF<x>_DALL:LABEL:FONT:ITALIC**

This command sets or queries the italic state of the specified digital group. The reference is specified by x.

**Group**  
Digital

**Syntax**  
REF<x>_DALL:LABEL:FONT:ITALIC {ON|OFF|<NR1>}

**Arguments**  
OFF argument turns off italic font.
ON argument turns on italic font.
<NR1> = 0 turns off italic font; any other value turns on italic font.
Examples

\texttt{REF1\_DALL:LABel:FONT:ITALic 1} turns on italic font.

\texttt{REF1\_DALL:LABel:FONT:ITALic?} might return
\texttt{:REF1\_DALL:LABEL:FONT:ITALIC 0} indicating the font is not italic.

\textbf{REF<x>_DALL:LABel:FONT:SIZE}

This command sets or queries the font size of the specified digital group. The reference is specified by \textit{x}.

\begin{itemize}
  \item **Group**: Digital
  \item **Syntax**: \texttt{REF<x>_DALL:LABel:FONT:SIZE <NR1>}
  \item **Arguments**: \texttt{<NR1>} is the font size.
  \item **Examples**: \texttt{REF1\_DALL:LABel:FONT:SIZE 16} sets the font size to 16 points.
  \texttt{REF1\_DALL:LABel:FONT:SIZE?} might return
  \texttt{:REF1\_DALL:LABEL:FONT:SIZE 20} indicating the font size is 20 points.
\end{itemize}

\textbf{REF<x>_DALL:LABel:FONT:TYPE}

This command sets or queries the font type of the specified digital group, such as Arial or Times New Roman. The reference is specified by \textit{x}.

\begin{itemize}
  \item **Group**: Digital
  \item **Syntax**: \texttt{REF<x>_DALL:LABel:FONT:TYPE <QString>}
  \item **Arguments**: \texttt{<QString>} is the font type.
  \item **Examples**: \texttt{REF1\_DALL:LABel:FONT:TYPE "Monosapce"} sets the font to a monospace font.
  \texttt{REF1\_DALL:LABel:FONT:TYPE?} might return
  \texttt{:REF1\_DALL:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman"}.
\end{itemize}
REF<x>_DALL:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified digital group. The reference is specified by x.

Group       Digital

Syntax      REF<x>_DALL:LABel:FONT:UNDERline {ON|OFF|<NR1>}

Arguments   OFF argument turns off underline font.
             ON argument turns on underline font.
             <NR1> = 0 turns off underline font; any other value turns on underline font.

Examples    REF1_DALL:LABel:FONT:UNDERline ON specifies an underlined font.
             REF1_DALL:LABel:FONT:UNDERline? might return :REF1_DALL:LABEL:FONT:UNDERLINE 0 indicating underline is off.

REF<x>_DALL:LABel:NAMe

This command sets or queries the label of the specified digital group. The reference is specified by x.

Group       Digital

Syntax      REF<x>_DALL:LABel:NAMe <QString>

Arguments   <QString> is the name of the group.

Examples    REF1_DALL:LABel:NAMe "Clock Out" sets the label name to Clock Out.
             REF1_DALL:LABel:NAMe? might return :REF1_DALL:LABEL:NAMe "This is the digital name".

REF<x>_DALL:LABel:XPOS

This command sets or queries the x-position of the label of the specified digital group. The reference is specified by x.
Commands listed in alphabetical order

REF<x>_DALL:LABEL:XPOS <NR3>

Arguments
<NR3> is the x-position, in pixels relative to the left edge of the display, of the group.

Examples
REF1_DALL:LABEL:XPOS 90.0e0 sets the x position of the label to 90.
REF1_DALL:LABEL:XPOS? might return :REF1_DALL:LABEL:XPOS 45.0000 indicating the x position of the label is at 45 pixels to the right of the left edge of the display.

REF<x>_DALL:LABEL:YPOS

This command sets or queries the y-position of the label of the specified digital group. The reference is specified by x.

Arguments
<NR3> is the y-position, in pixels relative to the baseline of the waveform, of the group.

Examples
REF1_DALL:LABEL:YPOS 50 sets the y position to 50.
REF1_DALL:LABEL:YPOS? might return :REF1_DALL:LABEL:YPOS 0.0E+0 indicating the y position of the label is at the baseline of the waveform.

REF<x>_D<x>:LABEL:COLOR

This command sets or queries the color of the label of the specified digital bit. The reference is specified by x.

Group  Digital
Syntax  REF<x>_D<x>:LABEL:COLOR <QString>
Arguments  <QString> is the label color. To return the color to the default color, send an empty string as in this example: :REF5_D1:LABEL:COLOR "".

Examples  REF1_D1:LABEL:COLOR "#FF0000" sets the color to red.
REF1_D1:LABEL:COLOR? might return :REF1_D1:LABEL:COLOR "#FFFF00" indicating the color is yellow.

**REF<x>_D<x>:LABel:FONT:BOLD**

This command sets or queries the bold state of the label of the specified digital bit. The reference is specified by x.

Group  Digital

Syntax  REF<x>_D<x>:LABel:FONT:BOLD {ON|OFF|<NR1>}

Arguments  OFF argument turns off bold font.
ON argument turns on bold font.
<NR1> = 0 turns off bold font; any other value turns on bold font.

Examples  REF1_D1:LABel:FONT:BOLD ON sets the font to bold.
REF1_D1:LABel:FONT:BOLD? might return :REF1_D1:LABEL:FONT:BOLD 0 indicating the font is not bold.

**REF<x>_D<x>:LABel:FONT:ITALic**

This command sets or queries the italic state of the label of the specified digital bit. The reference is specified by x.

Group  Digital

Syntax  REF<x>_D<x>:LABel:FONT:ITALiC {ON|OFF|<NR1>}

Arguments  OFF argument turns off italic font.
ON argument turns on italic font.
<NR1> = 0 turns off italic font; any other value turns on italic font.
Commands listed in alphabetical order

Examples  
REF1_D1:LABel:FONT:ITALic OFF turns off italic font.  
REF1_D1:LABel:FONT:ITALic? might return  
:REF1_D1:LABEL:FONT:ITALIC 1 indicating the font is italic.

REF<x>_D<x>:LABel:FONT:SIZE

This command sets or queries the font size of the label of the specified digital bit. The reference is specified by x.

Group  
Digital

Syntax  
REF<x>_D<x>:LABel:FONT:SIZE <NR1>

Arguments  
<NR1> is the font size.

Examples  
REF1_D1:LABel:FONT:SIZE 16 sets the font size to 16 points.  
REF1_D1:LABel:FONT:SIZE? might return :REF1_D1:LABEL:FONT:SIZE 20 indicating the font size is 20 points.

REF<x>_D<x>:LABel:FONT:TYPE

This command sets or queries the font type of the label of the specified digital bit, such as Arial or Times New Roman. The reference is specified by x.

Group  
Digital

Syntax  
REF<x>_D<x>:LABel:FONT:TYPE <QString>

Arguments  
<QString> is the font type of the label.

Examples  
REF1_D1:LABel:FONT:TYPE "Monospace" sets the font to Monospace.  
CHREF1_D1:LABel:FONT:TYPE? might return :REF1_D1:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman".
REF<x>_D<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the label of the specified digital bit. The reference is specified by x.

Group Digital

Syntax REF<x>_D<x>:LABel:FONT:UNDERline {ON|OFF|<NR1>}

Arguments OFF argument turns off underline font.
ON argument turns on underline font.
<NR1> = 0 turns off underline font; any other value turns on underline font.

Examples REF1_D1:LABel:FONT:UNDERline ON turns on underline font.
REF1_D1:LABel:FONT:UNDERline? might return :REF1_D1:LABEL:FONT:UNDERLINE 0 indicating the underline font is off.

REF<x>_D<x>:LABel:NAMe

Sets or queries the label of the specified digital bit. The channel is specified by x.

Group Digital

Syntax REF<x>_D<x>:LABel:NAMe <QString>

Arguments <QString> is the label.

Examples REF1_D1:LABel:NAMe "Clock in" sets the name to Clock in.
REF1_D1:LABel:NAMe? might return :REF1_D1:LABEL:NAMe "Digital 1".

REF<x>_D<x>:LABel:XPOS

This command sets or queries the x-position of the label of the specified digital bit. The reference is specified by x.

Group Digital
Commands listed in alphabetical order

**REF<x>_D<x>:LABel:XPOS <NR3>**

**Syntax**

REF<x>_D<x>:LABel:XPOS <NR3>

**Arguments**

<NR3> is the x-position, in pixels relative to the left edge of the display, of the label.

**Examples**

REF1_D1:LABel:XPOS 90 sets the x position to 90.

REF1_D1:LABEL:XPOS? might return :REF1_D1:LABEL:XPOS 45.0000 indicating the position is 45 pixels to the right of the left edge of the waveform.

**REF<x>_D<x>:LABel:YPOS**

This command sets or queries the y-position of the label of the specified digital bit. The channel is specified by x.

**Group**

Digital

**Syntax**

REF<x>_D<x>:LABel:YPOS <NR3>

**Arguments**

<NR3> is the y-position, in pixels relative to the baseline of the waveform, of the label.

**Examples**

REF1_D1:LABEL:YPOS 10 sets the y position to 10.

REF1_D1:LABEL:YPOS? might return :REF1_D1:LABEL:YPOS 0.0E+0 indicating the y position of the label is at the baseline of the waveform.

**REM (No Query Form)**

This command (no query form) embeds a comment within programs as a means of internally documenting the programs. This is how to embed comments in a .set file. The instrument ignores these embedded comment lines.

**Group**

Miscellaneous

**Syntax**

REM <QString>

**Arguments**

<QString> is a string that can contain a maximum of 80 characters.
Examples
REM "This is a comment" is a comment string that the instrument will ignore.

**ROSc:SOUrce**

This command sets or queries the selected source for the time base reference oscillator. The reference oscillator locks to this source. Depending on the command argument that you specify, you can use an external reference or use the internal crystal oscillator as the time base reference.

**Group**
Miscellaneous

**Syntax**
ROSc:SOURce {INTERnal|EXTernal}
ROSc:SOURce?

**Related Commands**
ROSc:STATE?

**Arguments**
INTERnal specifies the internal 10 MHz crystal oscillator as the time base reference.
EXTernal specifies the user-supplied external signal as the time base reference.

**Examples**
ROSC:SOURCe INTERNAL specifies the internal 10 MHz crystal oscillator as the time base reference.
ROSC:SOURCe? might return :ROSC:SOURCe INTERNAL, indicating that the 10 MHz crystal oscillator is being used as the time base reference.

**ROSc:STATE? (Query Only)**

This query-only command returns whether the time base reference oscillator is locked. This command will return either LOCKED or UNLOCKED.

**Group**
Miscellaneous

**Syntax**
ROSc:STATE?

**Returns**
LOCKED indicates the reference oscillator is locked.
UNLOCKED indicates the reference oscillator is not locked.
Examples

ROSC:STATE? might return :ROSC:STATE LOCKED, indicating that the time base reference is locked.

*RST (No Query Form)

This command (no query form) resets the instrument to the factory default settings.

This command does the following:

- Recalls the default instrument setup.
- Clears the current *DDT command.
- Disables aliases (:ALIAS:STATE 0).
- Disables the user password (for the *PUD command).
The *RST command does not change the following:

- The current working directory ("FILESystem:CWD command).
- The state of command headers ("HEADer command).
- The state of keyword and enumeration verbosity ("VERBose command).
- The Power-on Status Clear Flag (*PSC command).
- The Event Status Enable Register (*ESE command).
- The Service Request Enable Register (*SRE command).
- The Device Event Status Enable Register (DESE command).
- The user password for protected user data ("PASSword command).
- The content of protected user data (*PUD command).
- The enabled state of the socket server ("SOCKETServer:ENABLE command).
- The socket server port number ("SOCKETServer:PORT command).
- The socket server protocol ("SOCKETServer:PROTOCOL command).
- The USB TMC port configuration ("USBDevice:CONFigure command).
- The destination reference waveform or file path for the :CURVe command ("DATA:DESTination command).
- The waveform data encoding for the :CURVe command or query or the :WAVFrm? query ("DATA:ENCdg command).
- The starting point for :CURVe? queries ("DATA:STARt command).
- The ending point for :CURVe? queries ("DATA:STOP command).
- All settings associated the :WFMInpre commands.
- All user settable settings associated with the WFMOutpre commands.

*RST only resets the programmable interface settings, it does not change the user interface settings.

### Group

**Status and Error**

### Syntax

*RST

### Related Commands

FACTory
RECALL:SETUp
Arguments   None

Examples   *RST resets the instrument settings to factory defaults.

**SAVe:EVENTtable:BUS (No Query Form)**

This command saves bus results table to the specified file.

Group   Save and Recall

Syntax   SAVe:EVENTtable:BUS <QString>

Arguments   <QString> is the specified file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified.

- Complete path specification. If the file argument begins with drive designator (such as C’), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples   SAVe:EVENTtable:BUS "TEK000.CSV" saves the bus decode event table in the file named TEK000.CSV.

**SAVe:EVENTtable:CUSTom (No Query Form)**

This command saves the results table to the specified file path and name.

Group   Save and Recall

Syntax   SAVe:EVENTtable:CUSTom <QString>

Arguments   <QString> is the specified file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified.
Commands listed in alphabetical order

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

**Examples**

SAVE:EVENTtable:CUSTom "C:/TEK000.CSV" sets the oscilloscope to save the results table in the top level C: directory in a file named TEK000.CSV.

**SAVe:EVENTtable:CUSTom:COMMents**

This command sets or queries comments to be included in saved results table files.

**Group**
Save and Recall

**Syntax**

SAVe:EVENTtable:CUSTom:COMMeents <Qstring>

SAVe:EVENTtable:CUSTom:COMMeents?

**Arguments**

<Qstring> sets the oscilloscope to save the quoted string as a comment in the saved results table file.

**Examples**

SAVe:EVENTtable:CUSTom:COMMeents "PSRR batch 2 testing." sets the oscilloscope to save the quoted string in all following saved results table files.

SAVe:EVENTtable:CUSTom:COMMeents? might return "", indicating that there is no comment set to be saved in results table files.

**SAVe:EVENTtable:CUSTom:DATAFormat**

This command sets or queries the data format to use for saving results table data.

**Group**
Save and Recall

**Syntax**

SAVe:EVENTtable:CUSTom:DATAFormat [SCientific|ENGineering]

**Arguments**

SCientific sets the oscilloscope to save results tables data in scientific notation (for example, 5.0100E-12).
ENGINEering sets the oscilloscope to save results tables data in engineering notation (for example, 5.0100ps).

**Examples**

`SAVE:EVENTtable:CUSTom:DATAFormat SCIentific` sets the oscilloscope to save results tables data in scientific notation.

`SAVE:EVENTtable:CUSTom:DATAFormat SCIentific` might return `ENGINEERING`, indicating that the oscilloscope is set to save results tables data in engineering notation.

**SAVe:EVENTtable:CUSTom:INCLUDEREFs**

This command sets or queries whether to include displayed reference waveforms with saved results table files.

**Group**

Save and Recall

**Syntax**

`SAVe:EVENTtable:CUSTom:INCLUDEREFs {1|0}`

`SAVe:EVENTtable:CUSTom:INCLUDEREFs?`

**Arguments**

1 sets the oscilloscope to save all displayed reference waveforms as part of a saved results table file.

0 sets the oscilloscope to not save all displayed reference waveforms as part of a saved results table file.

**Examples**

`SAVe:EVENTtable:CUSTom:INCLUDEREFs 1` sets the oscilloscope to save all displayed reference waveforms as part of a saved results table file.

`SAVe:EVENTtable:CUSTom:INCLUDEREFs?` might return 0, indicating that the oscilloscope will not save all displayed reference waveforms as part of a saved results table file.

**SAVe:EVENTtable:MEASUrement (No Query Form)**

This command saves data (measurement) results to the specified file.

**Group**

Save and Recall

**Syntax**

`SAVe:EVENTtable:MEASUrement <QString>`
Arguments  

< QString > is the specified file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples  

SAVE:EVENTtable:MEASUrement "TEK000.CSV" saves the measurement in the file named TEK000.CSV.

SAVe:IMAGe (No Query Form)

Saves a capture of the screen contents to the specified image file. Supported image formats are PNG, Windows Bitmap, and JPEG.

Group  
Save and Recall

Syntax  
SAVe:IMAGe <QString>

Arguments  

< QString > is the file name and location used to store the image file.

When specifying the file name with this command, use the correct file extension (".png" for PNG format, "bmp" for BMP format, or ".jpg" for JPEG format). If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples  

SAVe:IMAGE “c:/rose_was_here.png” saves the image at the location specified.
SAVE:REPORT (No Query Form)

This command saves a report to the specified file. Supported report formats are PDF and MHT (web page archive file).

Group             Save and Recall
Syntax            SAVE:REPORT <QString>

Arguments
<QString> is the complete path specification. When specifying the file name with this command, use the correct file extension (.pdf for PDF format, or .mht for MHT format).

If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a Windows drive designator such as C:\, then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples
SAVE:REPORT "report.pdf" creates a report in PDF format, in the location specified.

SAVE:REPORT:COMMENTS

This command sets or queries the comments to be included in saved report files.

Group             Save and Recall
Syntax            SAVE:REPORT:COMMENTS <QString>
SAVE:REPORT:COMMENTS?

Arguments
<QString> is the comments to be included in saved report files.
Examples

SAVE:REPORT:COMMENTS "Test 3" adds comments to the report.


SAVE:SESSION (No Query Form)

Saves the state of the instrument, including reference waveforms, to a saved session file.

Group       Save and Recall

Syntax      SAVE:SESSION <QString>

Arguments   <QString> is the file path that specifies the location to save the specified instrument session file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWDFILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a Windows drive designator such as C:, then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or "." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples   SAVE:SESSION "c:/rose_was_here.tss" saves the instrument state in the specified file.

SAVE:SETUp (No Query Form)

Saves the current instrument state to the specified file.

Group       Save and Recall

Syntax      SAVE:SETUp <QString>

Related Commands   SAVE:SETUp:INCLUDEREFS
Commands listed in alphabetical order

Arguments

<QString> is a quoted string that is the complete path specification. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a Windows drive designator such as C:, then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ").." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

SAVE:SETUP "c:/rose_was_here.set" saves the instrument setup in the specified file.

SAVe:SETUp:INCLUDEREFs

This command sets or queries whether displayed reference waveforms are to be included in saved setups.

Group

Save and Recall

Syntax

SAVe:SETUp:INCLUDEREFs {OFF|ON|0|1}
SAVe:SETUp:INCLUDEREFs?

Arguments

OFF specifies not including displayed reference waveforms in saved setups.
ON specifies including displayed reference waveforms in saved setups.
0 specifies not including displayed reference waveforms in saved setups.
1 specifies including displayed reference waveforms in saved setups.

Examples

SAVE:SETUP:INCLUDEREFs 0 sets reference waveforms not to be included in saved setups.
SAVE:SETUP:INCLUDEREFs? might return :SAVE:SETUP:INCLUDEREFs 1 indicating that reference waveforms are to be included in saved setups.

SAVe:WAVEform (No Query Form)

This command saves the specified waveform(s) to the specified destination file. The waveform source or sources must be active (turned on) to save data to a file.
Group     Save and Recall

Syntax     SAVE:WAVEform \{CH<x>[\_DALL]|MATH<x>|REF<x>|ALL\},<QString>

Related Commands
FILESystem:CWD
SAVE:WAVEform SOURCELIst?

Arguments
<x> is the number of the analog channel, math, or reference waveform source used to save the waveform data.

_DALL saves the digital channel waveform data of the specified channel. This argument is required if the channel specified is a digital channel.

ALL saves all displayed analog, math, and reference waveforms to individual files. Each file name created includes the name of the source (ch1, math3, and so on) used to create that file.

<QString> is a quoted string that defines the path and file name to use to save the specified file, in the format ‘[<path>]<filename\.ext>’. Specifying a path is optional. If no path is entered, the file is saved to the current working directory as set in FILESystem:CWD.

<path> uses the form ‘<drive>/<dir>.../’. You can specify a relative path or a complete path:

- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a drive designator (such as C:), then the file name is interpreted as a full path from the specified drive.

<filename> sets the file name to use to create the file. A file can have up to 125 characters. When using the ALL argument to save multiple files, each filename has the filename appended with the source used to create that file. For example, a filename of QualTest can create QualTest_ch1.xxx, QualTest_ref1.xxx, and so on.

.<ext> sets the file format to which to save the data:

- Use the .wfm extension to save waveform data to a Tektronix Internal format.

- Use the .csv extension to save waveform data to a comma separated values spreadsheet format.
Examples  SAVE:WAVEFORM MATH1,"TEK0000.WFM" saves the Math1 waveform to the file TEK00000.WFM in the current working directory.

SAVE:WAVEform:SOURCELiSt? (Query Only)

This query returns a list of the available waveforms that can be specified as the source for the SAVE:WAVEform command. Source waveforms must have their display mode set to On to appear in this list and to be saved.

Group  Save and Recall

Syntax  SAVE:WAVEform:SOURCELiSt?

Examples  SAVE:WAVEform:SOURCELiSt? might return ALL, CH1_DALL, CH2, CH3, CH8, MATH1, MATH2, REF2, REF4.

SAVEON:FILE:DEST

This command sets or queries the location where files are saved when SAVEON:TRIGGER is ON and SAVEON:WAVEFORM is ON. You can save the files to a local drive or network path by entering the desired location in <QString>. You can also select to save the files to a USB drive.

Group  Save On

Syntax  SAVEON:FILE:DEST <QString>
        SAVEON:FILE:DEST?

Related Commands  SAVEON:FILE:NAME

Arguments  <QString> specifies the location to store files.

Examples  SAVEON:FILE:DEST "C:\users\username\Tektronix\TekScope\SaveOnTrigger" sets this as the location to save files (named by the SAVEON:FILE:NAME command), when there is a trigger.

SAVEON:FILE:DEST? might return :SAVEON:FILE:DEST "C:\users\username\Tektronix\TekScope\SaveOnEvent", indicating the drive location where files will be saved when there is a trigger.
SAVEON:FILE:NAME

Sets or queries the file name to use when SAVEON:TRIGer is ON.

Group  Save On

Syntax  SAVEON:FILE:NAME <QString>
        SAVEON:FILE:NAME?

Related Commands  SAVEON:FILE:DEST

Arguments  <QString> is the file name you want to use.

Examples  SAVEON:FILE:NAME "MaskFailure" sets the name of the file to MaskFailure.
        SAVEON:FILE:NAME? might return ":SAVEON:FILE:NAME MaskFailure5", indicating the name you set for the oscilloscope to use, with the autoincrement number (5) appended.

SAVEON:IMAGe

This command sets or queries whether to save a screen capture when a trigger occurs and SAVEON:TRIGer is ON and SAVEON:IMAGE is ON.

Group  Save On

Syntax  SAVEON:IMAGe {<NR1>|OFF|ON}
        SAVEON:IMAGe?

Related Commands  SAVEON:FILE:DEST
                    SAVEON:FILE:NAME
                    SAVEON:TRIGger
                    SAVEON:IMAGe

Arguments  <NR1> = 0 disables Save On Image; any other value turns this feature on.
           OFF disables Save On Image.
           ON enables Save On Image.
Examples

SAVEON:IMAGE ON sets the oscilloscope to save a screen capture on a specified trigger.

SAVEON:IMAGE? might return :SAVEON:IMAGE 1, indicating that the oscilloscope will save a screen capture when the specified trigger occurs.

SAVEON:IMAGE:FILEFormat

This command sets or queries the file format to be used for saved image files when :SAVEON:IMAGE is set to 1.

Group  Save On

Syntax

SAVEON:IMAGE:FILEFormat {PNG|BMP|JPG}
SAVEON:IMAGE:FILEFormat?

Arguments

PNG specifies using PNG format for saved image files.
BMP specifies using BMP format for saved image files.
JPG specifies using JPEG format for saved image files.

When specifying the file name with this command, use the correct file extension (".png" for PNG format, "bmp" for BMP format, or "jpg" for JPEG format). If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

SAVEON:IMAGE:FILEFORMAT JPG sets the image file format to JPEG.

SAVEON:IMAGE:FILEFORMAT? might return :SAVEON:IMAGE:FILEFORMAT PNG indicating that the file format is set to PNG.

SAVEON:TRIGger

Sets or queries whether to save a file when a trigger occurs. You can define the trigger using Trigger commands or the oscilloscope user interface.
The trigger will cause the instrument to save an image or a waveform to a file, depending on what you specified. For example, if you have set \texttt{SAVEON:IMAGe} to On, and a trigger event occurs, the instrument will save a screen capture. You can set options for file storage (such as file name, file destination, and auto increment), using the \texttt{SAVEON:FILE} commands.

Use the oscilloscope interface to select whether to save one or more analog channels, digital channels, or math waveforms.

Analog and math waveforms are saved using one file per waveform. Digital waveforms are all saved to a single file.

**Group**

- **Save On**

**Syntax**

\begin{verbatim}
SAVEON:TRIGger \{<NR1>|ON|OFF\}
SAVEON:TRIGger?
\end{verbatim}

**Related Commands**

- \texttt{SAVEON:IMAGe}
- \texttt{SAVEON:WAVEform}
- \texttt{SAVEON:FILE:DEST}
- \texttt{SAVEON:FILE:NAME}

**Arguments**

- \( <NR1> = 0 \) disables Save On Trigger, any other value turns this feature on.
- \texttt{OFF} disables Save On Trigger.
- \texttt{ON} enables Save On Trigger.

**Examples**

\begin{verbatim}
SAVEON:TRIGGER ON sets the oscilloscope to save an image, measurement, and/or waveform when a trigger occurs.
SAVEON:TRIGGER? might return :SAVEON:TRIGGER \ ON, indicating that a file will be saved on triggering.
\end{verbatim}

**SAVEON:WAVEform**

Sets or queries whether to save a waveform when a trigger occurs when \texttt{SAVEON:TRIGGER} is ON.

The waveform will be saved to the file you selected with \texttt{SAVEON:FILE:NAME}, in the location that you selected using \texttt{SAVEON:FILE:DEST}. You can set options for file storage (such as file name, file destination, and autoincrement), using the \texttt{SAVEON:FILE} commands.
Commands listed in alphabetical order

Group: Save On

Syntax:

SAVEON:WAVEform {<NR1>|ON|OFF}
SAVEON:WAVEform?

Related Commands:

SAVEON:FILE:DEST
SAVEON:FILE:NAME
SAVEON:TRIGger

Arguments:

<NR1> = 0 disables Save On Waveform, any other value turns this feature on.
OFF disables Save On Waveform.
ON enables Save On Waveform.

Examples:

SAVEON:WAVEFORM ON turns on the Save On Waveform feature, so that a waveform will be saved when a selected trigger occurs.
SAVEON:WAVEFORM? might return :SAVEON:WAVEFORM ON, indicating that a waveform will be saved when a selected trigger occurs.

SAVEON:WAVEform:FILEFormat

This command sets or queries the file format for saving waveforms when :SAVEON:WAVEform is set to 1.

Group: Save On

Syntax:

SAVEON:WAVEform:FILEFormat {INTERNal|SPREADSheet}
SAVEON:WAVEform:FILEFormat?

Arguments:

INTERNal specifies saving the waveform in the oscilloscope internal format.
SPREADSheet specifies saving the waveform in comma separated values format.

Examples:

SAVEON:WAVEFORM:FILEFORMAT SPREADSheet sets the file format to spreadsheet.
SAVEON:WAVEFORM:FILEFORMAT? might return :SAVEON:WAVEFORM:FILEFORMAT INTERNAL indicating the file format is set to INTERNAL.
SAVEON:WAVEform:SOURce

This command sets or queries the sources for saving waveforms when SAVEON:TRIGGER is ON.

Group: Save On

Syntax:  
SAVEON:WAVEform:SOURce \{CH<x>|MATH<x>|REF<x>|ALL\}  
SAVEON:WAVEform:SOURce?

Arguments:  
Arguments are the available sources.

Examples:  
SAVEON:WAVEform:SOURce MATH1 specifies MATH1 as the save on source.  
SAVEON:WAVEform:SOURce? might return :SAVEON:WAVEFORM:SOURCEREF1 indicating the save on source is REF1.

SEARCH:ADDNew (No Query Form)

This command adds the specified search.

Group: Search and Mark

Syntax:  
SEARCH:ADDNew <QString>

Arguments:  
<QString> is the specified search. The argument is of the form "SEARCH<NR1>", where <NR1> is \( \geq 1 \).

Examples:  
SEARCH:ADDNEW "SEARCH2" adds a new search named SEARCH 2.

SEARCH:DELETE (No Query Form)

This command deletes the specified search.

Group: Search and Mark

Syntax:  
SEARCH:DELETE <QString>
**Arguments**

<qstring> is the specified search. The argument is of the form "SEARCH<NR1>", where <NR1> is ≥ 1).

**Examples**

SEARCH:DELETE “SEARCH3” deletes SEARCH 3.

**SEARCH:LIST? (Query Only)**

This command returns a comma separated list of all currently defined searches.

**Group**

Search and Mark

**Syntax**

SEARCH:LIST?

**Returns**

All currently defined searches.

**Examples**

SEARCH:LIST? might return :SEARCH:LIST SEARCH1,SEARCH2 indicating that Search 1 and Search 2 are defined.

**SEARCH:SEARCH<x>::COPY (No Query Form)**

This command (no query form) copies the search criteria to or from the trigger. The search number is specified by <x>.

**Group**

Search and Mark

**Syntax**

SEARCH:SEARCH<x>::COPY {SEARCHtotrigger|TRIGgertosearch}

**Arguments**

SEARCHtotrigger copies the search criteria to the trigger.

TRIGgertosearch copies the trigger criteria to the search.

**Examples**

SEARCH:SEARCH1::COPY TRIGGERTOSEARCH copies the trigger criteria to the search 1 criteria.

SEARCH:SEARCH1::COPY SEARCHTOTRIGGER copies the search criteria to the trigger.
SEARCH:SEARCH<x>:NAVigate (No Query Form)

This command sets the navigation action for search marks. The NONE action is the default setting when no action is being taken. The search number is specified by <x>.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:NAVigate {NEXT|PREVIOUS|MIN|NONE|MAX}

Arguments NEXT goes to the next search mark.
PREVIOUS goes to the previous search mark.
MIN goes to the first search mark.
NONE is the default setting when no action is being taken.
MAX goes to the last search mark.

Examples SEARCH:SEARCH1:NAVigate NEXT goes to the next search mark.

SEARCH:SEARCH<x>:TOTAL? (Query Only)

This query-only command returns the total number of found search marks for this search. The search number is specified by <x>.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TOTAL?

Examples SEARCH:SEARCH1:TOTAL? might return SEARCH:SEARCH1:TOTAL 7, indicating that there are 7 matches for search 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:CONDition

This command specifies a field or condition for an ARINC429 bus to search on. The search number is specified by x.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.
Commands listed in alphabetical order

**Group**
Search and Mark

**Syntax**
{SOW|LABEL|DATA|LABELANDDATA|EOW|ERROR}

**Arguments**
- **SOW** specifies a search for the first bit of a word.
- **LABEL** specifies a search for a matching label.
- **DATA** specifies a search for matching packet data fields.
- **LABELANDDATA** specifies a search for a matching label and matching packet data field(s).
- **EOW** specifies a search for the 32nd bit of a word.
- **ERROR** specifies a search for a specified error condition.

**NOTE.** The type of error searched for is specified by

**Examples**
SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:CONDITION DATA specifies finding packets that contain matching data field(s).
SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:CONDITION? might return SOW, indicating that the bus is being searched for the first bit of each word in the packet.


This command sets or queries the high value when searching on an ARINC429 data field. The search number is specified by x. The search condition must be set to DATA or LABELANDDATA, and the data qualifier must be INrange or OUTrange.

**Conditions**
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
<QString>
Arguments  

<QString> is the label value.

Examples

"xxxxxxxxxxxxxxxxxx1000" sets the value to xxxxxxxxxxxxx1000.


This command sets or queries the qualifier to be used when searching on data in the DATA field for an ARINC429 bus signal. The search number is specified by x. The search condition must be set to DATA or LABELANDDATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGGER:A:BUS:ARINC429A:DATA:QUALIFIER {EQUAL|UNEQUAL|LESSThan|MOREThan |LESSEQUAL|MOREEQUAL|INrange|OUTrange}

Arguments

Arguments are the available data qualifiers.

NOTE. The search qualifier only applies to the bits defined as the data field via the bus data field format specifier (using BUS:B<x>:ARINC429A:DATABASE).

Examples


SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:DATA:QUALIFIER? might return EQUAL, indicating that the data qualifier is set to equal.


This command sets or queries the low value when searching on an ARINC429 data field. The search number is specified by x. The search condition must be set to DATA or LABELANDDATA.

This command sets or queries the error type when searching on an ARINC429 bus signal. The search number is specified by x. The search condition must be set to ERRor.

**Conditions**
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**

```
<QString>
```

**Arguments**

- `<QString>` is the label value.

**Examples**

```
SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:DATA:VALUE? might return "XXXXXXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXXXXXXXXX.
```

**SEARCH:SEARCH<x>:TRIGGER:A:BUS:ARINC429A:ERTYPE**

This command sets or queries the error type when searching on an ARINC429 bus signal. The search number is specified by x. The search condition must be set to ERRor.

**Conditions**
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**

```
{ANY|PARity|WORD|GAP}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:ARINC429A:ERTYPE?
```

**Arguments**

- **ANY** sets the error type to match any of the other available error types.
- **PARity** sets the error type to match on parity errors (parity value results in even parity count for a word).
- **WORD** sets the error type to match on word errors (any unframed or unknown decode data).
- **GAP** sets the error type to match on gap violations (less than 4 bits idle time between two packets on the bus).
**Examples**

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:ERRTYPE PARITY sets the error type to match on parity errors.


**SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:HIVALue**

This command sets or queries the high value when searching on an ARINC429 label field. The search number is specified by x. The search condition must be set to LABel, and the label qualifier must be INrange or OUTrange.

**Conditions**

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**

Search and Mark

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:HIVALue

<QString>

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:HIVALue?

**Arguments**

<QString> is the label value.

**Examples**


SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:HIVALUE? might return "XXXXXXXX", indicating that the value is XXXXXXXX.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:QUALi fier**

This command sets or queries the qualifier to be used when searching on label data for an ARINC429 bus signal. The search number is specified by x. The search condition must be set to LABel or LABELANDDATA.

**Conditions**

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**

Search and Mark
Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABEL:QUALifier {EQUAL|UNEQUAL|LESSthan|MORThan |
LESSEEQual|MOREEQual|INrange|OUTrange}

Arguments

Arguments are the available data qualifiers.

NOTE. If the search condition is set to LABELANDDATA, the label qualifier will be locked to Equal until the search condition is changed again.

Examples

LESSTHAN sets the label qualifier to less than.

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:QUALIFIER? might return EQUAL, indicating that the label qualifier is set to equal.


This command sets or queries the low value when searching on an ARINC429 label field. The search number is specified by x. The search condition must be set to LABEL or LABELANDDATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax


Arguments

<QString> is the label value.

Examples

"XXXX1010" sets the value to XXXX1010.

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:VALUE? might return "XXXXXXXX", indicating that the value is XXXXXXXX.

This command sets or queries the label when searching on an ARINC429 SDI field. The search number is specified by x. The search condition must be set to DATa or LABELANDDATA, and the data format must be set to DATA.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Search and Mark

Syntax

Arguments
<QString> is the label value.

NOTE. The SDI field is only present when the selected data field format is DATA (using BUS:B<x>:ARINC429A:DATAFORMAT). Also, the stored QString is reset to its default value whenever the data field format is changed.

Examples
SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:SDI:VALUE "X0" sets the value to X0.
SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:SDI:VALUE? might return "XX", indicating that the value is XX.


This command sets or queries the label when searching on an ARINC429 SSM field. The search number is specified by x. The search condition must be set to DATa or LABELANDDATA, and the data format must be set to DATa or SDDATA.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Search and Mark

Syntax
Arguments

<QString> is the label value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:SSM:VALUE "X0" sets the value to X0.

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:SSM:VALUE? might return "XX", indicating that the value is XX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:CONDition

This command sets or queries the condition (word select / frame sync, or matching data) to be used when searching on an audio bus signal. The search number is specified by <x>.

Conditions

Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:CONDition {SOF|DATA}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:CONDition?

Arguments

SOF specifies to search on a word select or start of frame (depending on Audio Type).

DATA specifies to search on matching data.

Examples


This command sets or queries the binary data string for the high data word to be used when searching on an TDM audio bus signal. The search condition must be set to DATa using SEARCH:SEARCH{x}:TRIGger:A:BUS:AUDio:CONDition.

Group

Search and Mark
Commands listed in alphabetical order


Syntax

```
<QString>
```

Arguments

<QString> is the binary data string for the high data word to be used when searching on an TDM audio bus signal.

Examples

"XXXXXXXXXXXXXXXXXXXX1100" sets the high value to 1100.

"XXXXXXXXXXXXXXXXXXXX1010" indicating the high value is set to 1010.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HIVALue**

This command sets or queries the binary data string for the high data word to be used when searching on an audio bus signal. The search condition must be set to DATa using :SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:CONDition. The search number is specified by <x>.

Group

Search and Mark

Syntax

```
```

Arguments

<QString> specifies the upper word value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:HIVALue “XXXX” sets the HIVALUE to XXXX.

"TEST_001101”, indicating the HIVALUE is set to TEST_001101.


This command sets or queries the data offset value (TDM channel) to be used when searching on a TDM type audio bus signal. The search condition must be set to DATa using SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:TDM:CONDition. The search number is specified by <x>.
Commands listed in alphabetical order

**Conditions**
Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDIO:DATA:OFFSET <NR1>
```

**Arguments**
`<NR1>` is the data offset value.

**Examples**

SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:OFFSET 1 indicating data offset value is set to TDM channel 1.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDIO:DATA:QUALifier**

This command sets or queries the qualifier to be used when searching on an audio bus signal. The search condition must be set to DATa using SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDIO:{NONTdm|TDM}:CONDITION. The search number is specified by `<x>`.

**Conditions**
Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDIO:DATA:QUALifier {EQual|UNEQual|LESSthan|MOREthan|LESSEQual|MORREQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDIO:DATA:QUALifier?
```

**Arguments**
`LESSthan` sets the qualifier to less than.

`MOREthan` sets the qualifier to greater than.

`EQual` sets the qualifier to equal.

`UNEQual` sets the qualifier to not equal.

`LESSEQual` sets the qualifier to less than or equal.
MOREEQual sets the qualifier to greater than or equal.
INrange sets the qualifier to in range.
OUTrange sets the qualifier to out of range.

**Examples**


**SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:TDMVALue**

This command sets or queries the binary data string for the single or low data word to be used when searching on an TDM audio bus signal. The search condition must be set to DATA using SEARCH:SEARCH{x}:TRIGger:A:BUS:AUDIO:CONDition.

**Group** Search and Mark

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:TDMVALue?

**Arguments**

<QString> is the binary data string for the single or low data word to be used when searching on an TDM audio bus signal.

**Examples**


**SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:VALue**

This command sets or queries the binary data string for the single or low data word to be used when searching on an audio bus signal. The search condition must be set to DATA using SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDIO:{NONTdm|TDM}:CONDition. The search number is specified by <x>.
Conditions Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group Search and Mark

Syntax


Arguments <QString> is the lower word value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:VALUE "X0X011" sets the data value to X0X011.


This command sets or queries the alignment of the data (left, right or either) to be used when searching on a non-TDM type audio bus signal. The search condition must be set to DATA using SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:NONTdm:CONDition. The search number is specified by <x>.

Conditions Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:WORD {EITHER|LEFT|RIGHT}

Arguments EITHER aligns the data to either left or right.

LEFT aligns the data to the left.

RIGHT aligns the data to the right.
Commands listed in alphabetical order

Examples


SEARCH:SEARCH<x>:TRIGGER:A:BUS:CAN:CONDITION

This command sets or queries the search condition for a CAN bus. The search number is specified by <x>.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGGER:A:BUS:CAN:CONDITION {SOF|FRAMEtype|IDENTifier|DATA|IDANDDATA|EOF|ERROR|FDBITS}

SEARCH:SEARCH<x>:TRIGGER:A:BUS:CAN:CONDITION?

Arguments

Arguments specify the CAN bus trigger condition.

Examples


This command specifies the CAN search type to be valid on a Read, Write, or Either condition. The search number is specified by <x>.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGGER:A:BUS:CAN:DATA:DIRection {READ|WRITE|NOCARE}

Arguments

READ specifies the read direction.
WRITE specifies the write direction.
NOCARE specifies either data direction.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATA:DIRection READ sets the data direction to READ.
SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:DIRection NOCARE, indicating the data direction is set to either data direction.


This command sets or queries the data offset value, in bytes, to use when searching on the CAN data field. The search number is specified by x. The search condition must be set to DATA or IDANDDATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:OFFSet <NR1>

Related Commands

BUS:B<x>:CAN:STANDard
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:VALue

Arguments

<NR1> is an integer whose minimum and default values are -1 (don't care) and maximum is up to 7 (for CAN 2.0) or up to 63 (for ISO CAN FD and Non-ISO CAN FD). The maximum is dependent on the number of bytes being matched and the CAN standard selected. Its value is calculated as [Absolute Maximum] - [Data Match Size]. For CAN 2.0, the absolute maximum is 8 bytes. For ISO CAN FD and Non-ISO CAN FD, the absolute maximum is 64 bytes. The minimum data match size is 1 byte, which produces the ranges listed above. Increasing the data match size above 1 byte will adjust the range of valid data offset values accordingly.
Examples

sets the CAN data offset to 5 bytes.

SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:OFFSET? might return 7, indicating the CAN data offset is 7 bytes. If the CAN standard is set for CAN 2.0 and the search data size is set to 3, the maximum value for the data offset will be 5 \((8 - 3 = 5)\). If the CAN standard is set for ISO CAN FD or Non-ISO CAN FD and the search data size is set to 8, the maximum value for the data offset will be 56 \((64 - 8 = 56)\).


This command sets or queries the CAN bus trigger data qualifier to be used when searching on a CAN bus signal. The search number is specified by \(<x>\).

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**

```
{EQUAL|LESS_EQUAL|MORE_EQUAL|UNEQUAL|LESS_THAN|MORE_THAN}
```

**Arguments**
Arguments are the data qualifier types.

**Examples**

sets the data qualifier to unequal.


This command sets or queries the length of the data string, in bytes, to be used when searching on a CAN bus signal. The search condition must be set to IDANDDATA or DATA. The search number is specified by \(<x>\).

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.
Commands listed in alphabetical order

Group: Search and Mark

Syntax:

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:SIZE <NR1>

Arguments:

<NR1> specifies the data size.

Examples:

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATA:SIZE 1, indicating the data size is set to 1.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:VALue**

This command sets or queries the binary data value to be used when searching on a CAN bus signal. The search condition must be set to IDANDDATA OR DATA.

Group: Search and Mark

Syntax:

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:VALue <QString>

Arguments:

<QString>

Examples:

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATA:VALue "1111" sets the data value to 1111.
SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATA:VALue "1010" indicating the data value is 1010.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType**

This command sets or queries the type of error condition for a CAN bus to search on. The search number is specified by x. The search condition must be set to ERRor.

Conditions:

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.
Commands listed in alphabetical order

Group       Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGGER:A:BUS:CAN:ERType
{ACKMISS|BITSTUFFing|FORMERRor |ANYERRor}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:CAN:ERType?

Arguments

ACKMISS specifies a search based on a missing ACK field.
BITSTUFFing specifies a search based on a bit stuffing error.
FORMERRor specifies a search based on a CAN FD form error. To use this option, the CAN standard must be set to FDISO or FDNONISO.
ANYERRor specifies a search based on any error type.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:ERTYPE? might return ANYERROR, indicating that the bus is being searched for all error types.


This command sets or queries the value of the bit rate switch bit (BRS bit) for a CAN bus to search on. The search number is specified by x. The search condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group       Search and Mark

Syntax

{ONE|ZERO|NOCARE}

Arguments

ONE filters CAN FD packets to only match those where the BRS bit has a value of 1 (fast data enabled).
ZERO filters CAN FD packets to only match those where the BRS bit has a value of 0 (fast data disabled).
NOCARE disables filtering of CAN FD packets on the BRS bit.
Examples

```
```


This command sets or queries the value of the error state indicator bit (ESI bit) for a CAN bus to search on. The search number is specified by x. The search condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

**Conditions**

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**

Search and Mark

**Syntax**

```
{ONE|ZERO|NOCARE}
```

**Arguments**

ONE filters CAN FD packets to only match those where the ESI bit has a value of 1 (recessive).

ZERO filters CAN FD packets to only match those where the ESI bit has a value of 0 (dominant).

NOCARE disables filtering of CAN FD packets on the ESI bit.

**Examples**

```
```

**SEARCH:SEARCH<x>:TRIGGER:A:BUS:CAN:FRAMEtype**

This command sets or queries CAN bus trigger frame type to be used when searching on a CAN bus signal. The search condition must be set to FRAMEtype. The search number is specified by <x>.
**SEARCH:**

**SEARCH<x>:TRIGger:A:BUS:CAN:FRAMEtype {DATA|ERROR|OVERLoad|REMOTE}**

- **Arguments**: Arguments are the available frame types.

- **Examples**:
  - `SEARCH:SEARCH1:TRIGger:A:BUS:CAN:FRAMEtype ERROR` sets the frame type to error.

**SEARCH:**

**SEARCH<x>:TRIGger:A:BUS:CAN:IDentiﬁer:MODE {EXTENDed|STandard}**

- **Conditions**: Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

- **Group**: Search and Mark

- **Syntax**:

- **Arguments**:
  - EXTENDED specifies the extended identifier mode.
  - STANDARD specifies the standard identifier mode.

- **Examples**:
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDENTifier:VALue

This command sets or queries CAN bus trigger identifier (address) value to be used when searching on a CAN bus signal. The search number is specified by <x>. The search condition must be set to IDANDDATA or DATA.

Conditions  Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDENTifier:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDENTifier:VALue?

Arguments  <QString> is the identifier value.

Examples  SEARCH:SEARCH1:TRIGger:A:BUS:CAN:IDENTifier:VALue "1010" sets the identifier value to 1010.
indicating the identifier value is 101011.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

This command specifies a field or condition for an Ethernet bus to search on. The search number is specified by <x>.

Conditions  Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax  SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition
{SFD|MACADDRess|MACLENgt|IPHeader|TCPHeader|DATa|EOP|
Commands listed in alphabetical order

```
IDLe|FCSError|QTAG
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDITION?

**Related Commands**  
Most of the other TRIGger:A:BUS:B<x>:ETHERnet commands are impacted by the setting of this command.

**Arguments**  
- **SFD**  
  - Start of frame delimiter.
- **MACADDRESS**  
  - MAC addresses field.
- **MACLENGTH**  
  - MAC length/type field.
- **IPHeader**  
  - IP header field.
- **TCPHeader**  
  - TCP header field.
- **DATA**  
  - TCP/IPv4 or MAC protocol client data field.
- **EOP**  
  - End of Packet field.
- **IDLe**  
  - Idle field.
- **FCSError**  
  - Frame Check Sequence Error (CRC) field.
- **QTAG**  
  - IEEE 802.1Q (VLAN) control information field.

**Examples**  
SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:CONDITION MACADDRESS 

specifies MACADDRESS as the field within an Ethernet frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:CONDITION? might return DATA, indicating that DATA is the currently specified field within an Ethernet frame to search on.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:HIVALue**

This command sets or queries the binary data value to be used when searching on an Ethernet bus signal. The search condition must be set to DATA and the data qualifier to inside or outside range. The search number is specified by <x>.

**Conditions**  
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**  
Search and Mark

**Syntax**  
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:HIVALue <QString>
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:HIVALue?

Arguments

<QString> is the binary data value to be used when searching on an Ethernet bus signal.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:ETHERnet:DATa:HIVALue "101011" sets the data HIVALue to 101011.


This command specifies the data offset value, in bytes, to use when searching on the Ethernet data field. The search condition needs to be set to DATa. The search number is specified by <x>.

Conditions

Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:OFFSet <NR1>

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet CONDITION
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet DATa:VALUE
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet DATa:SIZe

Arguments

<NR1> is an integer whose minimum and default values are -1 (don't care) and maximum is 1,499.

Examples


SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:OFFSET? might return -1, indicating that the data offset value is the default value, -1, meaning "don't care".

This command specifies the qualifier to be used when searching on an Ethernet bus signal. The search condition must be set to DATa. The search number is specified by <x>.

**Conditions**
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
{LESSthan|MOREthan|EQUAL|UNEQUAL|LESSEEQUAL|MOREEQUAL|INrange|OUTrange}


**Related Commands**
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

**Arguments**
LESSthan sets the qualifier to less than.
MOREthan sets the qualifier to greater than.
EQUAL sets the qualifier to equal.
UNEQUAL sets the qualifier to not equal.
LESSEEQUAL sets the qualifier to less than or equal.
MOREEQUAL sets the qualifier to greater than or equal.
INrange sets the qualifier to in range.
OUTrange sets the qualifier to out of range.

**Examples**
LESSTHAN sets the qualifier to "less than".


This command specifies the length of the data string, in bytes, to use when searching on the Ethernet bus signal. The search condition needs to be set to DATa. The search number is specified by <x>.

**Conditions**
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZE <NR1>

**Related Commands**
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

**Arguments**
The minimum and default values are 1, and the maximum value is 16.

**Examples**


SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:VALue

This command specifies the binary value to use when searching on the Ethernet bus signal. The search condition needs to be set to DATa. The search number is specified by <x>.

**Conditions**
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:VALue?
**Related Commands**


**Arguments**

- `<QString>` is a quoted string where the allowable characters are 0, 1, and X. The allowable number of characters depends on the setting for size (using `SEARCH:SEARCH<x>:TRIGGER:A:BUS:ETHERnet:DATA:SIZE`). The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

**Examples**


This command specifies the binary destination address value to use when searching on an Ethernet bus signal. The search condition needs to be set to `IPHeader`. The search number is specified by `<x>`.

**Conditions**

Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**

Search and Mark

**Syntax**

```
```

**Related Commands**


**Arguments**

- `<QString>` is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.
**Examples**

```
SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:IPHEADER:
DESTINATIONADDR:VALUE "11000000101010000000000000000001"
```

specifies 192.168.0.1 as the value to use when searching on the Ethernet IPv4 header address destination field.

```
SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:IPHEADER:
DESTINATIONADDR:VALUE? might return
"110000001010100000000000000000001", indicating that 192.168.0.1 is
the currently specified value used when searching on the Ethernet IPv4 header address destination field.
```


This command specifies the binary protocol value to use when searching on the Ethernet bus signal. The search condition needs to be set to IPHeader. The search number is specified by <x>.

---

**NOTE.** Commonly used protocol values are 1 (ICMP), 2 (IGMP), 6 (TCP) and 17 (UDP).

---

**Conditions**

Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**

Search and Mark

**Syntax**

```
```

**Related Commands**

SEARCH:SEARCH<x>:TRIGGER:A BUS:ETHERnet CONDITION

**Arguments**

<QString> is a quoted string of up to 8 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

**Examples**

```
"01010010" specifies 01010010 as the value to use when searching on the Ethernet IP header protocol field.
```
Commands listed in alphabetical order


This command specifies the binary source address value to use when searching on an Ethernet bus signal. The search condition needs to be set to IPHeader. The search number is specified by <x>.

**Conditions** Requires option 5-SREN or SUP5-SREN Triggering and Analysis application.

**Group** Search and Mark

**Syntax**

```
```

**Related Commands** 
SEARCH:SEARCH<x>:TRIGGER:A:BUS:ETHERnet:CONDition

**Arguments**

QString is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

**Examples**

```
```

```
SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERnet:IPHeader:SOURceaddr:VALUE might return "11000000101010000000000000000001", indicating that the search value has been set to 192.168.0.1.
```


This command specifies the binary MAC address destination value to use when searching on an Ethernet bus signal. The search condition needs to be set to MACADDRess. The search number is specified by <x>.

**NOTE.** MAC Addresses are 48-bit values such as 08:00:11:1E:C9:AE hex.
Commands listed in alphabetical order

**Conditions**
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**

**Related Commands**
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

**Arguments**
<QString> is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

**Examples**


This command specifies the binary MAC address source value to use when searching on an Ethernet bus signal. The search condition needs to be set to MACADDRess. The search number is specified by <x>.

**NOTE.** MAC Addresses are 48-bit values such as 08:00:11:1E:C9:AE hex.

**Conditions**
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**
Search and Mark
Commands listed in alphabetical order

Syntax


Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments

<QString> is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples


This command specifies the binary MAC length high value to use when searching on an Ethernet bus signal. The search condition needs to be set to MACADDresS. The search number is specified by <x>.

Conditions

Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group

Search and Mark

Syntax


Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments

<QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.
Examples

"xxxxxxxxxxxx00001000" specifies to use the hexadecimal value XX08 when searching on the Ethernet MAC length.

might return "xxxxxxxxxxxx00001000", indicating an Ethernet MAC length value of XX08 hex.

This command specifies the MAC length value to use when searching on an Ethernet bus signal. The search condition needs to be set to MACADDRESS. The search number is specified by <x>.

Conditions
Requires option 5-SRENETH or SUP5-SRENETH Triggering and Analysis application.

Group
Search and Mark

Syntax
QString

Related Commands
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments
QString is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

"xxxxxxxxxxxx00001000" specifies to use the hexadecimal value XX08 when searching on the Ethernet MAC length.

might return "xxxxxxxxxxxx00001000", indicating an Ethernet MAC length value of XX08 hex.

This command specifies the binary Q-tag information to use when searching on an Ethernet bus signal. The search condition needs to be set to QTAG. The search number is specified by <x>.
Commands listed in alphabetical order

Conditions
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group
Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:QTAG:VALue?
```

Related Commands
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition
```

Arguments
<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples
```
SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:QTAG:VALUE? might return "XXXXXXXXXXXXXXXXXXXX010010001010", indicating that hexadecimal XXXXXXX8A has been set as the Ethernet Q-Tag field search value.
```


This command specifies the binary ack number value to use when searching on an Ethernet bus signal. The default is all X's (don't care). The search condition needs to be set to TCPHeader. The search number is specified by <x>.

Conditions
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group
Search and Mark

Syntax
```
```

Related Commands
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition
```
Arguments

QString is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:ACKNUM:VALUE "XXXXXXXXXXXXXXXXXXXX00001000" specifies hexadecimal XXXXXX08 as the value to use when searching on the Ethernet TCP header acknowledgement number.

TRIGGER:A:BUS:ETHERNET:TCPHEADER:ACKNUM:VALUE? might return "XXXXXXXXXXXXXXXXXXXX00001000", indicating that hexadecimal XXXXXX08 has been specified as the value to use when searching on the Ethernet TCP header acknowledgement number.


This command specifies the binary destination port value to use when searching on the Ethernet TCP header destination port number. The search condition needs to be set to TCPHeader. The search number is specified by <x>.

Conditions

Requires option 5-SREN or SUP5-SREN Triggering and Analysis application.

Group

Search and Mark

Syntax


Related Commands


Arguments

<QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

Commands listed in alphabetical order

SEARCH: SEARCH<x>: TRIGGER:A: BUS: ETHERNET: TCPHEADER: DESTINATIONPORT: VALUE? might return "XXXXXXXXXXXXXXXX", indicating that hexadecimal XXXX has been set as the value to use when searching on the Ethernet TCP header destination port number.


This command specifies the binary sequence number value to use when searching on an Ethernet bus signal. The default is all X’s (don’t care). The search condition needs to be set to TCPHeader. The search number is specified by <x>.

**Conditions**
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**

**Related Commands**

**Arguments**
QString is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

**Examples**

SEARCH: SEARCH1: TRIGGER:A: BUS: ETHERNET: TCPHEADER: SEQNUM: VALUE? might return "XXXXXXXXXXXXXXXXXXXXXX0010010100", indicating that hexadecimal XXXX111 has been specified as the value to use when searching on the Ethernet TCP header sequence number.


This command specifies the binary source port value to use when searching on an Ethernet bus signal. The search condition needs to be set to TCPHeader. The search number is specified by <x>.
Commands listed in alphabetical order

**Conditions**
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
```
```

**Related Commands**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition
```

**Arguments**
<(QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

**Examples**
```
```

```
SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:SOURCEPORT:VALUE? might return "XXXXX01001010110", indicating that hexadecimal X0A6 has been specified as the value to use when searching on the Ethernet TCP header source port number.
```

**SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CONDition**

This command sets or queries the FlexRay bus search condition. The search number is specified by <x>.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CONDition {SOF|FRAMEType|IDentifier|CYCLEcount|HEADer|DATA|IDANDDATA|EOF|ERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CONDition?
```

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer 2-823
Arguments  Arguments are the available trigger conditions.

Examples  SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:CONDITION SOF sets the trigger condition to start of frame.


**SEARCH:SEARCH<x>:TRIGGER:A:BUS:FLEXRAY:CYCLEcount:HIVALue**

This command sets or queries the heigh value when searching on a FlexRay bus cycle count field. The search number is specified by <x>. The search condition must be set to CYCLEcount.

Conditions  Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group  Search and Mark


<QString>

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:HIVALue?

Arguments  <QString> specifies the cycle count value.


"1010" sets the cycle count value to XX1010.


might return


"XXXXXX", indicating the cycle count value is don’t care, and it will trigger on any cycle count.

**SEARCH:SEARCH<x>:TRIGGER:A:BUS:FLEXRAY:CYCLEcount:QUALifier**

This command sets or queries the qualifier to be used when searching on a FlexRay bus search cycle count field. The search number is specified by <x>. The search condition must be set to CYCLEcount.
Commands listed in alphabetical order

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier
{EQUAL|LESSEQUAL|MOREEQUAL|UNEQUAL|LESSthan|MOREthan|
INrange|OUTrange}
```
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier?
```

**Arguments**
Arguments are the available cycle count qualifiers.

**Examples**
```
SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier LESSthan sets the cycle count qualifier to less than.
```
```
SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier?
might return
SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLECOUNT:QUALIFIER EQUAL, indicating the cycle count qualifier is set to equal.
```

**SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue**

This command sets or queries the low value when searching on a FlexRay bus cycle count field. The search number is specified by <x>. The search condition must be set to CYCLEcount.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue
<QString>
```
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue?
```

**Arguments**
<QString> specifies the cycle count value.

**Examples**
```
SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue "1010"
sets the cycle count value to XX1010.
```


This command sets or queries the high value when searching on a FlexRay bus data field. The search number is specified by <x>. The search condition must be set to IDANDDATA OR DATA.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark


Arguments <QString> specifies the data value.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:DATA:HIVALUE "1010" sets the data value to XXXXXX1010.


This command sets or queries the offset of the data string, in bytes, when searching on a FlexRay bus data field. The search number is specified by <x>. The search condition must be set to IDANDDATA OR DATA.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark
Commands listed in alphabetical order

**SEARCH:**

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Arguments</th>
<th>Examples</th>
</tr>
</thead>
</table>

**SEARCH:**

This command sets or queries the qualifier to be used when searching on a FlexRay bus signal. The search number is specified by <x>.

**Conditions**

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**

Search and Mark

**Syntax**

```plaintext
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATA:QUALifier

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATA:QUALifier {EQUal|LESSEQual|MOREEQual|UNEQual|LESSthan|MOREthan|INrange|OUTrange}
```

**Arguments**

Arguments are the available data qualifiers.

**Examples**

```plaintext
SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATA:QUALifier LESSTHAN sets the data qualifier to less than.

```
Commands listed in alphabetical order

**SEARCH:**SEARCH<\textit{x}>:**TRIGger:**A:**BUS:**FLEXRAY:**DATa:**SIZe

This command sets or queries the length of the data string, in bytes, to be used when searching on a FlexRay bus data field. The search number is specified by \textit{x}. The search condition must be set to IDANDDATA OR DATA.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
SEARCH:**SEARCH<\textit{x}>:**TRIGger:**A:**BUS:**FLEXRAY:**DATa:**SIZe  \texttt{<NR1>}
SEARCH:**SEARCH<\textit{x}>:**TRIGger:**A:**BUS:**FLEXRAY:**DATa:**SIZe? 

**Arguments**
\texttt{<NR1>} specifies the data size in bytes. A data size of –1 is don’t care.

**Examples**
SEARCH:**SEARCH1:**TRIGger:**A:**BUS:**FLEXRAY:**DATa:**SIZe 2 sets the data size to 2.
SEARCH:**SEARCH1:**TRIGger:**A:**BUS:**FLEXRAY:**DATa:**SIZe? might return SEARCH:**SEARCH1:**TRIGger:**A:**BUS:**FLEXRAY:**DATa:**SIZe 1, indicating the data size is 1 byte.

**SEARCH:**SEARCH<\textit{x}>:**TRIGger:**A:**BUS:**FLEXRAY:**DATa:**VALue

This command sets or queries the low value to be used when searching on a FlexRay bus data field. The search number is specified by \textit{x}. The search condition must be set to IDANDDATA or DATA.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
SEARCH:**SEARCH<\textit{x}>:**TRIGger:**A:**BUS:**FLEXRAY:**DATa:**VALue  \texttt{<QString>}
SEARCH:**SEARCH<\textit{x}>:**TRIGger:**A:**BUS:**FLEXRAY:**DATa:**VALue? 

**Arguments**
\texttt{<QString>} specifies the data value.
Commands listed in alphabetical order

**Examples**

```
SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:DATA:VALUE "1010" sets the data value to XXXXXXX1010.
```

**SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:EOFTYPE**

This command sets or queries the end of frame type when searching on a FlexRay bus signal. The search number is specified by <x>. The search condition must be set to EOF.

**Conditions**

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**

Search and Mark

**Syntax**

```
SEARCH:SEARCH<x>:TRIGGER:A:BUS:FLEXRAY:EOFTYPE {ANY|STATIC|DYNAMIC}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:FLEXRAY:EOFTYPE?
```

**Arguments**

ALL specifies either end of file type.

STATIC specifies the static end of file type.

DYNAMIC specifies the static end of file type.

**Examples**

```
```

**SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:ERRTYPE**

This command sets or queries the error type when searching on a FlexRay bus signal. The search number is specified by <x>. The search condition must be set to ERROR.

**Conditions**

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.
Group: Search and Mark

Syntax:

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:ERRTYPE
{CRCHeader|CRCTrailer|NULLFRStatic|NULLFRDynamic|SYNCFrame|STARTupnosync}
```

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:ERRTYPE?
```

Arguments: Arguments are the available error types.

Examples:

```
SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:ERRTYPE SYNCFRAME sets the error type to SYNCFRAME.
```

```
```

**SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:HIVALue**

This command sets or queries the high value when searching on a FlexRay bus frame id field. The search condition must be set to IDentifier. The search number is specified by <x>.

Group: Search and Mark

Syntax:

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:HIVALue
QString
```

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:HIVALue?
```

Arguments: <QString> is the frame id high value.

Examples:

```
SEARCH:SEARCH1:TRIGger:A:BUS:FLEXray:FRAMEID:HIVALue "XXXXXXX1010" sets the HIVALUE to XXXXXXX1010.
```

```
```

**SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:QUALi**

This command sets the qualifier to be used when searching on a FlexRay bus signal. The search condition must be set to IDentifier. The search number is specified by <x>.

2-830 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
Commands listed in alphabetical order

Group: Search and Mark

Syntax:

{EQUAL|UNEQUAL|LESSthan|MOREthan|LESSEQUAL|MOREEQUAL|
INrange|OUTrange}

Arguments:

Arguments are the available data qualifiers.

Examples:

UNEQUAL sets the qualifier to unequal.

EQUAL indicating the qualifier is set to equal.


This command sets the low value when searching on a FlexRay bus id field. The
search condition must be set to IDentity. The search number is specified by <x>.

Group: Search and Mark

Syntax:

<QString>

Arguments:

<QString> is the frame id value.

Examples:

"XXXXXXXX1010" sets the value to XXXXXXXX1010.

"XXXXXXXXXXXX" indicating the value is XXXXXXXXXXXX

SEARCH:SEARCH<x>:TRIGGER:A:BUS:FLEXray:FRAMEType

This command sets or queries the FlexRay bus search frame type. The search
number is specified by <x>.
Commands listed in alphabetical order

**GROUP**

**SEARCH and Mark**

**Syntax**

```
SEARCH:SEARCH<x>:TRIGGER:A:BUS:FLEXRAY:FRAMEType {NORMAL|PAYLOAD|NULL|SYNC|STARTup}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:FLEXRAY:FRAMEType?
```

**Arguments**
Arguments are the available frame types.

**Examples**

```
SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:FRAMEType startup sets the frame type to startup.
```


This command sets or queries the CRC portion of the binary header string when searching on a FlexRay bus header. The search number is specified by `<x>`.

**Syntax**

```
```

**Arguments**

`<QString>` specifies the CRC.

**Examples**

```
```

This command sets or queries the cycle count portion of the binary header string when searching on a FlexRay bus header. The search number is specified by <x>. The search condition must be set to HEADer.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CYCLECount?

**Arguments**
<QString> specifies the header cycle count.

**Examples**
SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:HEADER:CYCLECount "1010" sets the header cycle count to XX1010.


This command sets or queries the frame id portion of the binary header string when searching on a FlexRay bus header. The search number is specified by <x>. The search condition must be set to HEADer or IDANDDATA.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
<QString>

**Arguments**
<QString> specifies the frame ID.
Example

```
SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:HEADER:FRAMEID "1010"
sets the header frame ID to XXXXXXXX1010.

indicating the frame ID is a don’t care.
```

**SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:INDBits**

This command sets or queries the Indicator bits portion of the binary header string when searching on a FlexRay bus header. The search number is specified by <x>. The search condition must be set to HEADer.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
```
<QString>
```

**Arguments**

- `<QString>` specifies the header Indicator Bits.

**Examples**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:INDBits "1010"
sets the header Indicator Bits to 1010.

indicating the Indicator Bits have not been set.
```

**SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:PAYLENGth**

This command sets or queries the payload length portion of the binary header string when searching on a FlexRay bus search trigger header. The search number is specified by <x>. The search condition must be set to HEADer.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Search and Mark
Commands listed in alphabetical order


Syntax

```
<QString>
```

Arguments

<QString> specifies the header Payload Length.

Examples

```
sets the Payload Length to 1010.

SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:HEADER:PAYLENGTH "xxxxxxxx", indicating the Payload Length has not been set.
```

**SEARCH:SEARCH<x>:TRIGGER:A:BUS:I2C:ADDRESS:MODE**

This command sets or queries the I2C address mode for the specified bus search to determine where to place a mark. The search number is specified by <x>.

Syntax

```
{ADDR10|ADDR7}
```

Arguments

ADDR10 specifies the address mode as ADDR10.

ADDR7 specifies the address mode as ADDR7.

Examples

```

SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:ADDRESS:MODE ADDR7, indicating that the address mode for I2C bus trigger search 2 is set to ADDR7.
```

This command sets or queries the binary address string used for the I2C search. The specified search condition is Address or AddressData. The search number is specified by <x>.

Conditions
Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group
Search and Mark

Syntax

Related Commands

Arguments
<QString> specifies the address value. This is either a 7-bit or 10-bit value depending on the address mode. The valid characters are 0-9, A-F, and X for addresses in hexadecimal format; and 0, 1, and X otherwise.

Examples
SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:ADDRESS:VALUE "01XXXXX" sets the address value to "01XXXXX" when the mode is ADDR7 and the format is binary.
SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:ADDRESS:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:ADDRESS:VALUE "XX", indicating that the address value is "XX" when the address mode is set to ADDR7 and the address format is hexadecimal.

SEARCH:SEARCH<x>:TRIGGER:A:BUS:I2C:CONDITION

This command sets or queries the search condition for an I2C bus. The search number is specified by <x>.

Conditions
Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGGER:A:BUS:I2C:CONDITION
{ADDRESS|ADDREADWRITE|DATA|ACKMISS|REPEATstart|START|STOP}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:I2C:CONDITION?

Arguments

ADDRESS specifies the trigger condition as Address.
ADDRANDDATA specifies the trigger condition as Address and Data.
DATA specifies the trigger condition as Data.
ACKMISS specifies the trigger condition as Missing of Acknowledgement.
REPEATAStart specifies the trigger condition as Repeat of Start.
START specifies the trigger condition as Start.
STOP specifies the trigger condition as Stop.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:CONDITION ADDRESS sets the trigger condition for I2C bus trigger search 1 to Address.


This command sets or queries the direction of the data for the I2C bus search to determine where to place a mark. The search number is specified by <x>. Read or write is indicated by the R/W bit in the I2C protocol.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGGER:A:BUS:I2C:DATA:DIRection {NOCARE|READ|WRITE}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:I2C:DATA:DIRection?

Arguments

NOCARE specifies the direction of data as Don’t Care.
READ specifies the direction of data as Read.
WRITE specifies the direction of data as Write.
Examples

SEARCH: SEARCH1: TRIGGER:A: BUS: I2C: DATA: DIRECTION READ sets the data direction for I2C bus trigger search 1 to READ.


This command sets or queries the length of the data string in bytes used for an I2C bus search to determine where to place a mark. The search number is specified by <x>. The search condition must be DATA or ADDRANDDATA.

**Conditions**

Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

**Group**

Search and Mark

**Syntax**

SEARCH: SEARCH<x>: TRIGGER:A: BUS: I2C: DATA: SIZE <NR1>

**Arguments**

<NR1> specifies the data size in bytes.

**Examples**

SEARCH: SEARCH1: TRIGGER:A: BUS: I2C: DATA: SIZE 1 sets the length of the data string for I2C bus trigger search 1 to 1 byte.


This command sets or queries the binary data string used for I2C bus search to determine where to place a mark. The search number is specified by <x>. The search condition must be DATA or ADDRANDDATA.

**Conditions**

Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

**Group**

Search and Mark
Commands listed in alphabetical order

Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:VALue?

Arguments
<QString> specifies the data value. The valid characters are 0, 1, or X for binary format; and A-F, 0-9, and X for hexadecimal format.

Examples
SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:VALUE "1001" sets the data value for I2C bus trigger search 1 to "1001".

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition

This command sets or queries the condition for a LIN bus search. The search number is specified by <x>.

Conditions
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition {DATA|IDANDDATA|ERRror|IDentifier|SLEEP|SYNCfield|WAKEup}
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition?

Arguments
Arguments are the available trigger conditions.

Examples
SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:CONDition DATA sets the trigger condition to data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:HIVALue

This command sets or queries the high data value string used in a LIN bus search. The search number is specified by <x>. The search condition must be DATA or IDANDDATA and the data qualifier must be INRANGE or OUTRANGE.
Commands listed in alphabetical order

**GROUP**

**Search and Mark**

**Syntax**


**Arguments**

<QString> is a quoted string of 1s, 0s, or Xs representing the binary data string to be used in a LIN search if the search condition is IDentifier or IDANDDATA (identifier and data).

**Examples**


This command sets or queries the data qualifier used in a LIN bus search. The search number is specified by <x>.

**Conditions**

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**GROUP**

**Search and Mark**

**Syntax**

SEARCH:SEARCH<x>:TRIGGER:A:BUS:LIN:DATA:QUALIFIER {EQUAL|LESSEEQUAL|MOREEQUAL|UNEQUAL|LESSTHAN|MORETHAN|INRANGE|OUTRANGE}


**Arguments**

Arguments are the available data qualifiers.

**Examples**


This command sets or queries the length of the stat string in bytes used for a LIN bus search. The search number is specified by <x>.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:SIZE <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:SIZE?

Arguments <NR1> specifies the data size.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:LIN:DATa:SIZE 1.0 sets the data size to 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:VALue

This command sets or queries the data string used for a LIN bus search. The search number is specified by <x>. The search condition must be DATA or IDANDDATA.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:VALue?

Arguments <QString> specifies the data value.

Examples SEARCH:SEARCH2:TRIGger:A:BUS:LIN:DATa:VALue 1010 sets the data value to XXXXXXX1010.
Commands listed in alphabetical order

indicating that the data value is a don’t care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE

This command sets or queries the error type for a LIN bus search. The search number is specified by <x>. The search condition must be set to ERROR.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE
{Checksum|Parity|SYNC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE?

**Arguments**
Checksum specifies the error type is checksum.
Parity specifies the error type is parity.
SYNC specifies the error type is sync.

**Examples**
:SEARCH:SEARCH2:TRIGGER:A:BUS:LIN:ERRTYPE SYNC, indicating that the error type is sync.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:IDenti fi er:VALue

This command sets or queries the string used for a LIN bus identifier value. The search number is specified by <x>. The search condition must be IDENTIFIER or IDANDDATA.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Search and Mark
Commands listed in alphabetical order

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:IDentifier:VALue

Arguments

<QString> specifies the identifier value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:LIN:IDentifier:VALue 1010 sets the identifier value to XX1010.


This command sets or queries the high value when searching on command word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to COMMAND, and the address qualifier must be INRange or OUTrange.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax


Arguments

<QString> is the address value.

Examples


SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:ADDRESS:HIVALUE? might return "XXXXXX", indicating that the value is XXXXX.

This command sets or queries the qualifier to be used when searching on command word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to COMMAND.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRESS:QUALifier {EQUAL|UNEQUAL|LESSthan |MOREthan|LESSEQual|MOREEQual|INrange|OUTrange}

Arguments
Arguments are the available address qualifiers.

Examples
LESSTHAN sets the address qualifier to less than.
might return EQUAL, indicating that the address qualifier is set to equal.


This command sets or queries the low value when searching on command word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to COMMAND.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Search and Mark

Syntax
Commands listed in alphabetical order

**Arguments**

<Qstring> is the address value.

**Examples**


SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:ADDRESS:VALUE? might return "XXXXX", indicating that the value is XXXXX.


This command sets or queries the value of the command word "word count" field for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to COMMAND.

**Conditions**

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**

Search and Mark

**Syntax**

SEARCH:SEARCH<x>:TRIGGER:A:BUS:MIL1553B:COMMAND:COUNT <Qstring>

**Arguments**

<Qstring> is the word count value.

**Examples**

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:COUNT "X1000" sets the value to X1000.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:COUNT? might return "XXXXX", indicating that the value is XXXXX.

**SEARCH:SEARCH<x>:TRIGGER:A:BUS:MIL1553B:COMMAND:PARity**

This command sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to COMMAND.

**Conditions**

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.
Group: Search and Mark

Syntax:

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:PARity
{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:PARity?

Arguments:

ONE filters command words to only match those where the parity bit has a value of 1.

ZERO filters command words to only match those where the parity bit has a value of 0.

NOCARE disables filtering of command words on the parity bit.

Examples:

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:PARITY ONE specifies filtering command words for those where the parity bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:PARITY? might return NOCARE, indicating that command words are not being filtered based on the parity bit value.


This command sets or queries the value of the command word subaddress field for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to COMMAND.

Conditions:
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group: Search and Mark

Syntax:

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:SUBADdress <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:SUBADdress?

Arguments:

<QString> is the word count value.

Examples:

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:SUBADDRESS "X1000" sets the value to X1000.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:SUBADDRESS? might return "XXXXX", indicating that the value is XXXXX.
**SEARCH:**SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:TRBit

This command sets or queries the value of the command word Transmit / Receive bit for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to COMMAND.

**Conditions** Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group** Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:TRBit
{RX|TX|x}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:TRBit?
```

**Arguments**
- RX filters command words to only match those that are receive packets.
- TX filters command words to only match those that are transmit packets.
- x disables filtering of command words on the R/T bit.

**Examples**
```
specifies filtering command words for only transmit messages.
```

**SEARCH:**SEARCH<x>:TRIGger:A:BUS:MIL1553B:CONDition

This command sets or queries the field or condition for a MIL-STD-1553 bus to search on. The search number is specified by x.

**Conditions** Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group** Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:CONDition
{SYNC|COMMAND|STATUS|DATA|ERROR}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:CONDition?
```

**Group** Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:CONDition
{SYNC|COMMAND|STATUS|DATA|ERROR}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:CONDition?
```
Arguments

SYNC specifies a search for the sync pulse of any word.
COMMAND specifies a search for a matching command word.
STATUS specifies a search for a matching status word.
DATA specifies a search for a matching data word.
ERROR specifies a search for a specified error condition.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:CONDITION DATA specifies finding matching data word(s).
SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:CONDITION? might return SYNC, indicating that the bus is being searched for sync pulses found in any word.

SEARCH:SEARCH<x>:TRIGGER:A:BUS:MIL1553B:DATA:PARity

This command sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to DATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGGER:A:BUS:MIL1553B:DATA:PARity
{ONE|ZERO|NOCARE}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:MIL1553B:DATA:PARity?

Arguments

ONE filters data words to only match those where the parity bit has a value of 1.
ZERO filters data words to only match those where the parity bit has a value of 0.
NOCARE disables filtering of data words on the parity bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:DATA:PARITY ONE specifies filtering data words for those where the parity bit has a value of 1.
SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:DATA:PARITY? might return NOCARE, indicating that data words are not being filtered based on the parity bit value.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:VALue

This command sets or queries the value when searching on data words for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to DATA.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:VALue?

Arguments
<QString> is the data value.

Examples
SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:DATA:VALUE "XXXXXXXXXXXX1000" sets the value to XXXXXXXXXXXX1000.
SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:DATA:VALUE? might return "XXXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:ERRTYPe

This command sets or queries the type of error condition for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to ERROR.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:ERRTYPe {PARity|SYNC|DATA}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:ERRTYPe?
Arguments

**PARity** specifies searching for an incorrectly calculated parity bit in any word.

**SYNC** specifies searching for any sync pulse that does not transition in the middle of the pulse as required.

**DATA** specifies searching for any non-contiguous data words.

Examples


SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:ERRTYPE? might return PARITY, indicating that the bus is being searched for parity errors in any word.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:HIVALue**

This command sets or queries the high value when searching on status word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to STATus and the address qualifier must be INrange or OUTrange.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax


Arguments

<QString> is the address value.

Examples


SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATus:ADDRess:HIVALUE? might return "XXXXX", indicating that the value is XXXXX.


This command sets or queries the qualifier used when searching on status word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to STATus.
Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Search and Mark

Syntax
{EQUAL|UNEQUAL|LESSthan|MOREthan|LESSEEQUAL|MOREEQUAL|INrange|OUTrange}

Arguments
Arguments are the available address qualifiers.

Examples
LESS THAN sets the address qualifier to less than.
might return EQUAL, indicating that the address qualifier is set to equal.

This command sets or queries the low value when searching on status word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to STATus.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Search and Mark

Syntax
<QString>

Arguments
<QString> is the address value.

Examples
"X1000" sets the value to X1000.
might return "XXXXX", indicating that the value is XXXXX.


This command sets or queries the value of the broadcast command received bit (BCR bit, bit 15) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

**Conditions** Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group** Search and Mark

**Syntax**
```
{ONE|ZERO|NOCARE}
```

**Arguments**
- **ONE** filters status words to only match those where the BCR bit has a value of 1.
- **ZERO** filters status words to only match those where the BCR bit has a value of 0.
- **NOCARE** disables filtering of status words on the BCR bit.

**Examples**
```
SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:BCR? might return NOCARE, indicating that status words are not being filtered based on the BCR bit value.
```


This command sets or queries the value of the busy bit (BUSY bit, bit 16) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

**Conditions** Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group** Search and Mark
Commands listed in alphabetical order

**Syntax**

```
{ONE|ZERO|NOCARE}
```

**Arguments**

- **ONE** filters status words to only match those where the BUSY bit has a value of 1.
- **ZERO** filters status words to only match those where the BUSY bit has a value of 0.
- **NOCARE** disables filtering of status words on the BUSY bit.

**Examples**

```
```

specifies filtering status words for those where the BUSY bit has a value of 1.

```
```

might return NOCARE, indicating that status words are not being filtered based on the BUSY bit value.


This command sets or queries the value of the dynamic bus control acceptance bit (DBCA bit, bit 18) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

**Conditions**

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**

Search and Mark

**Syntax**

```
{ONE|ZERO|NOCARE}
```

**Arguments**

- **ONE** filters status words to only match those where the DBCA bit has a value of 1.
- **ZERO** filters status words to only match those where the DBCA bit has a value of 0.
- **NOCARE** disables filtering of status words on the DBCA bit.

**Examples**

```
```

specifies filtering status words for those where the DBCA bit has a value of 1.

```
```

might return NOCARE, indicating that status words are not being filtered based on the DBCA bit value.

This command sets or queries the value of the instrumentation bit (INSTR bit, bit 10) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

**Conditions**
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
{ONE | ZERO | NOCARE}

**Arguments**
ONE filters status words to only match those where the INSTR bit has a value of 1.
ZERO filters status words to only match those where the INSTR bit has a value of 0.
NOCARE disables filtering of status words on the INSTR bit.

**Examples**
SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATus:BIT:INSTR? might return NOCARE, indicating that status words are not being filtered based on the INSTR bit value.


This command sets or queries the value of the message error bit (ME bit, bit 9) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

**Conditions**
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
{ONE | ZERO | NOCARE}
Arguments

ONE filters status words to only match those where the ME bit has a value of 1.
ZERO filters status words to only match those where the ME bit has a value of 0.
NOCARE disables filtering of status words on the ME bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:ME ONE specifies filtering status words for those where the ME bit has a value of 1.
SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:ME? might return NOCARE, indicating that status words are not being filtered based on the ME bit value.


This command sets or queries the value of the status word service request bit (SRQ bit, bit 11) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

{ONE|ZERO|NOCARE}

Arguments

ONE filters status words to only match those where the SRQ bit has a value of 1.
ZERO filters status words to only match those where the SRQ bit has a value of 0.
NOCARE disables filtering of status words on the SRQ bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:SRQ ONE specifies filtering status words to those where the SRQ bit has a value of 1.
SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:SRQ? might return NOCARE, indicating that status words are not being filtered based on the SRQ bit value.

This command sets or queries the value of the subsystem flag bit (SUBSF bit, bit 17) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x>. The search condition must be set to STATus.

**Conditions**
Requires SR-AERO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
{ONE|ZERO|NOCARE}

**Arguments**
ONE filters status words to only match those where the SUBSF bit has a value of 1.
ZERO filters status words to only match those where the SUBSF bit has a value of 0.
NOCARE disables filtering of status words on the SUBSF bit.

**Examples**


This command sets or queries the value of the terminal flag bit (TF bit, bit 19) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x>. The search condition must be set to STATus.

**Conditions**
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
{ONE|ZERO|NOCARE}
Arguments

- **ONE** filters status words to only match those where the TF bit has a value of 1.
- **ZERO** filters status words to only match those where the TF bit has a value of 0.
- **NOCARE** disables filtering of status words on the TF bit.

Examples


```
```

This command sets or queries the value of the status word parity bit for a MIL-STD-1553 bus to search on. The search number is specified by `x`. The search condition must be set to `STATUS`.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
{ONE|ZERO|NOCARE}
```

Examples

- `SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:PARITY?` might return NOCARE, indicating that status words are not being filtered based on the parity bit value.

This command sets or queries the binary data string used for a parallel bus search to determine where to place a mark. The search number is specified by `<x>`.

**Group** Search and Mark

**Syntax**

```
```

**Arguments**

- `<QString>` specifies the data value in a valid format. Valid characters are 0-9.

**Examples**

```
sets the string data value to "XXXXXXXX" in binary format.

returns SEARCH:SEARCH1:TRIGGER:A:BUS:PARALLEL:DATA:VALUE "XXXXXXXXXXXXXXX10010", indicating that the data value string is set to "XXXXXXXXXXXXXXX10010" in binary format.
```

**SEARCH:SEARCH<x>:TRIGGER:A:BUS:RS232C:CONDITION**

This command sets or queries the condition for an RS232C bus search to determine where to place a mark. The search number is specified by `<x>`.

**Conditions** Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

**Group** Search and Mark

**Syntax**

```
SEARCH:SEARCH<x>:TRIGGER:A:BUS:RS232C:CONDITION {DATA|EOP|PARITYERROR|START}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:RS232C:CONDITION?
```

**Arguments**

- `DATA` specifies the search condition as Data.
- `EOP` specifies the search condition as End of Packet.
- `PARITYERROR` specifies the search condition as Parity Error.
- `START` specifies the search condition as Start.
Examples


This command sets or queries the length of the data string in bytes to be used for an RS232 bus search to determine where to place a mark when the search condition is Data. The search number is specified by <x>.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Search and Mark

Syntax


Arguments

<NR3> is the number of bits per word in the data string, from 1 to 8.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:DATA:SIZe 1 sets the number of bits per word in RS232C bus trigger search 1 data string to 1.


This command sets or queries the data string used for the specified RS232C bus trigger search to determine where to place a mark. The search condition must be Data. The search number is specified by <x>.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Search and Mark
Commands listed in alphabetical order

Syntax


Arguments

<QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format, and A-F, 0-9, and X for values in hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:DATA:VALUE "01" sets the value of the data string for RS232C bus trigger search 1 to "01" when the format is hexadecimal.


SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

This command sets or queries the search condition for a SENT bus.

Conditions

Requires option SRAUTOSEN.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition {START|FAST|SLOW|PAUSE|ERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition?

Arguments

Search<x> is the number of the search.

START specifies searching for start of packet.

FAST specifies searching for fast channel data.

SLOW specifies searching for slow channel data.

PAUSE specifies searching for pause pulses.

ERROR specifies searching on errors.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:CONDITION PAUSE, indicating 
the search is set to find pause pulses.

SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:ERRType

This command sets or queries the error type to be used when searching on SENT data.

**Conditions**
Requires option SRAUTOSEN.

SENT bus trigger condition is set to ERRor.

**Group**
Search and Mark

**Syntax**
SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:ERRType 
{FRAMELENGTH|CRC}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:ERRType?

**Related Commands**
SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:CONDITION

**Arguments**
Search<x> is the number of the search.

FRAMELENGTH specifies searching for SENT frame length errors.

CRC specifies searching for CRC errors.

**Examples**
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:ERRTYPE FRAMELENGTH sets the 
search to find SENT frame length errors.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:ERRTYPE CRC to indicate the 
search is set to find CRC errors.


This command sets or queries the CRC error type to be used when searching 
on SENT data.

**Conditions**
Requires option SRAUTOSEN.

SENT bus trigger condition is set to ERRor and ERRType is set to CRC.
Group: Search and Mark

Syntax:
- `SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRC {FAST|SLOW}`
- `SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRC?`

Related Commands:
- `SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType`

Arguments:
- `Search<x>` is the number of the search.
- `FAST` specifies searching for CRC errors only in the fast channel.
- `SLOW` specifies searching for CRC errors only in the slow channel.

Examples:

**SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue**

This command sets or queries the high binary fast channel 1 value to use when searching on a SENT bus signal.

Conditions:
- Requires option SRAUTOSEN.
- The SENT bus search condition must be set to FAST.

Group: Trigger

Syntax:
- `SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue?`

Related Commands:

Arguments:
- `Search<x>` is the Search identifier number.
- `<QString>` sets the Fast Channel 1 binary data high value.
Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:HIVALue "XXXXXXXXXXXXX" sets the Fast Channel 1 high value on which to search to XXXXXXXXXXXXX, or "don't care."


SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:QUALifier

This command sets or queries the qualifier to be used when searching on SENT fast packet bus data for device channel 1.

Conditions

Requires option SRAUTOSEN

The SENT bus trigger condition must be set to FAST.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:QUALifier {EQual|UNEQual|LESSthan|MOREthan|LESSEQual|MOREEQual|INrange|OUTrange}

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:QUALifier?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue

Arguments

Search<x> is the number of the search.

EQUal specifies the qualifier as Equal.

LESSEQual specifies the qualifier as Less Than or Equal to.

LESSThan specifies the qualifier as Less Than.

MOREEQual specifies the qualifier as More Than or Equal to.

MOREThan specifies the qualifier as More Than.

UNEQual specifies the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.
Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:QUALIFIER UNEQUAL sets the fast channel 1 data qualifier to not equal for search 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:QUALIFIER MOREEQUAL to indicate that the fast channel 1 data qualifier is set to greater than or equal for search 3.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue**

This command sets or queries the binary fast channel 1 value to be used when searching on a SENT bus signal.

**Conditions**

Requires option SRAUTOSEN.

The trigger condition must be set to FAST.

**Group**

Search and Mark

**Syntax**

<Qstring>
SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:FAST:CHAN1A:VALUE?

**Related Commands**

SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:CONDition

**Arguments**

<x> is the number of the search.

<Qstring> is the Fast Channel 1 binary value.

**Examples**

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:VALUE "XXXXXXXXXXXX" sets the Fast Channel 1 value to XXXXXXXXXXXX, or "don't care."


**SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:HIVALue**

This command sets or queries the high binary fast channel 2 value to use when searching on a SENT bus signal.
**Commands listed in alphabetical order**

**Conditions**
Requires option SRAUTOSEN.
The SENT bus search condition must be set to FAST.

**Group**
Trigger

**Syntax**
```
<QString>
```

**Related Commands**
`SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition`

**Arguments**
`Search<x>` is the Search identifier number.

`<QString>` sets the Fast Channel 2 high binary data value.

**Examples**
```
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:HIVALue "100000000000" sets the Fast Channel 2 high value on which to search to 100000000000.

```

### SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:QUALifier

This command sets or queries the qualifier to be used when searching on SENT fast packet bus data for device channel 2.

**Conditions**
Requires option SRAUTOSEN.
The SENT bus trigger condition must be set to FAST.

**Group**
Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:QUALifier {Equal|UNEQual|LESSthan|MOREthan|LESSEEQual|MOREEQual|INrange|OUTrange}
```

**Related Commands**
`SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition`

---

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer 2-865

**Arguments**

- Search<x> is the number of the search.
- EQUAL specifies the qualifier as Equal.
- LESSEQUAL specifies the qualifier as Less Than or Equal to.
- LESS_THAN specifies the qualifier as Less Than.
- MOREEQUAL specifies the qualifier as More Than or Equal to.
- MORE_THAN specifies the qualifier as More Than.
-UNEQUAL specifies the qualifier as Unequal.

**Examples**


This command sets or queries the binary fast channel 2 value to be used when searching on a SENT bus signal.

**Conditions**

- Requires option SRAUTOSEN.
- The trigger condition must be set to FAST.

**Group**

Search and Mark

**Syntax**

```
Qstring
```

**Related Commands**

SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:CONDITION
Arguments

Search<x> is the number of the search.

<string> is the Fast Channel 2 binary value.

Examples

"111111111111" sets the Fast Channel 2 value to 111111111111.

"000000000000" to indicate the binary value 000000000000.


This command sets or queries the high binary fast message counter value to use when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.
The SENT bus search condition must be set to FAST.
The number of channels must be set to 1.
The nibble count must be set to 6.

Group

Trigger

Syntax


Related Commands

SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:CONDition
BUS:B<x>:SENT:NUMCHANnel
BUS:B<x>:SENT:NIBBLECount

Arguments

Search<x> is the Search identifier number.

<string> sets the Fast Channel 1 counter binary value.

Examples

"xxxxxxxxxx" sets the Fast Channel secure counter high value on which to search to "don't care."

This command sets or queries the qualifier to be used when searching on SENT fast packet bus data for the secure format counter.

**Conditions**
Requires option SRAUTOSEN

The SENT bus trigger condition must be set to FAST.

**Group**
Search and Mark

**Syntax**
```
LESSEqual|MOREEqual|INrange|OUTrange}
```

**Related Commands**

**Arguments**
- `Search<x>` is the number of the search.
- `EQUal` specifies the qualifier as Equal.
- `LESSEqual` specifies the qualifier as Less Than or Equal to.
- `LESSthan` specifies the qualifier as Less Than.
- `MOREEqual` specifies the qualifier as More Than or Equal to.
- `MOREthan` specifies the qualifier as More Than.
- `UNEQual` specifies the qualifier as Unequal.
- `INrange` sets the qualifier to inside a range.
- `OUTrange` sets the qualifier to outside a range.

**Examples**
- `SEARCH:SEARCH5:TRIGGER:A:BUS:SENT:FAST:COUNTER:QUALifier LESSthan` sets the fast channel 2 data qualifier on search 5 to less than.
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:QUALIFIER EQUAL to indicate that the fast channel 2 data qualifier on search 1 is set to equal.


This command sets or queries the binary fast message counter value to be used when searching on a SENT bus signal.

Conditions
- Requires option SRAUTOSEN.
- The search condition must be set to FAST.
- The number of channels must be set to 1.
- The nibble count must be set to 6.

Group
Search and Mark

Syntax

Related Commands
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition
BUS:B<x>:SENT:NIBBLECount
BUS:B<x>:SENT:NUMCHANnel

Arguments
Search<x> is the number of the search.
<Qstring> is the Fast Channel 1 counter value.

Examples
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:VALUE "XXXXXXXX" sets the Fast Channel 1 secure counter value to "don't care."


This command sets or queries the binary fast message inverted nibble value to be used when searching on a SENT bus signal.
Commands listed in alphabetical order

**Conditions**
Requires option SRAUTOSEN.
The search condition must be set to FAST.
The number of channels must be set to 1.
The nibble count must be set to 6.

**Group**
Search and Mark

**Syntax**

**Related Commands**
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition
BUS B<x>:SENT:NIBBLECount
BUS B<x>:SENT:NUMCHANnel

**Arguments**
Search<x> is the number of the search.
<Qstring> is the fast message inverted nibble binary value.

**Examples**
TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE "XXXX" sets the Fast Channel 1 secure inverted nibble value to "don't care."


This command sets or queries the binary status value to be used when searching on a SENT bus signal.

**Conditions**
Requires option SRAUTOSEN.
The trigger condition must be set to FAST.

**Group**
Search and Mark

**Syntax**
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:STATus:VALue <Qstring>
Commands listed in alphabetical order


Related Commands
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDITION

Arguments
Search<x> is the number of the search.
Qstring is the binary status binary value.

Examples
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:STATUS:VALUE "XXXX" sets the binary status value to XXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:QUALifier

This command sets or queries the qualifier to be used when searching on SENT pause pulses.

Conditions
Requires option SRAUTOSEN
The SENT bus trigger condition must be set to PAUSE.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:QUALifier {EQUAL|UNEQUAL|LESSthan|MOREthan|LESSEqual|MOREEqual|INrange|OUTrange}

Related Commands
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDITION

Arguments
Search<x> is the number of the search.
EQUAL specifies the qualifier as Equal.
INrange sets the qualifier to be within a range.
LESSEqual sets the qualifier as Less Than or Equal to.
LESSThan sets the qualifier as Less Than.
MOREEqual sets the qualifier as More Than or Equal to.
MOREThan sets the qualifier as More Than.
OUTRange sets the qualifier to be outside a range.
UNEQual specifies the qualifier as Unequal.

Examples

This command sets or queries the maximum number of pause clock ticks to be used when searching on a SENT bus signal.

Conditions
Requires option SRAUTOSEN.
The trigger condition must be set to PAUSE.

Group
Search and Mark

Syntax

Related Commands
SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:CONDition

Arguments
Search<x> is the number of the search.
<NRL> is the maximum number of pause clock ticks to be used when searching.

Examples

This command sets or queries the minimum number of pause clock ticks to be used when searching on a SENT bus signal.

**Conditions**
Requires option SRAUTOSEN.
The trigger condition must be set to PAUSE.

**Group**
Search and Mark

**Syntax**
SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:PAUSE:TICKs:VALUE <NR1>

**Related Commands**
SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:CONDITION

**Arguments**
Search<x> is the number of the search.

<NR1> is the minimum number of pause clock ticks to be used when searching.

**Examples**
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKs:VALUE 12 sets the minimum pause tick count on which to search to 12.


This command sets or queries the high binary Slow channel data value to use when searching on a SENT bus signal.

**Conditions**
Requires option SRAUTOSEN.
The SENT bus search condition must be set to SLOW.

**Group**
Trigger

**Syntax**
Related Commands

SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:CONDITION

Arguments

search<x> is the Search identifier number.

Examples

"XXXXXXXX" sets the Slow data high value on which to search to "don't care" for Search 2.

"11110001", to indicate searching on slow data values that match 11110001.


This command sets or queries the qualifier to be used when searching on SENT slow packet bus data.

Conditions

Requires option SRAUTOSEN.
The SENT bus trigger condition must be set to SLOW.

Group

Search and Mark

Syntax

LESSEQUAL|MOREEQUAL|INrange|OUTrange}

Related Commands

SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:CONDITION

Arguments

search<x> is the number of the search.

EQUAL specifies the qualifier as Equal.

LESSEQUAL sets the qualifier as Less Than or Equal to.

LESSThan sets the qualifier as Less Than.

MOREEQUAL sets the qualifier as More Than or Equal to.

MOREThan sets the qualifier as More Than.
Commands listed in alphabetical order

UNEQual specifies the qualifier as Unequal.

INrange sets the search qualifier to inside a range.

OUTrange sets the search qualifier to outside a range.

Examples

```
```

LESSEQUAL sets the slow channel data qualifier to less than or equal.

```
```

might return

```
```

EQUAL to indicate that the slow channel data qualifier is set to equal.


This command sets or queries the binary slow channel data value to be used when searching on a SENT bus signal.

**Conditions**

Requires option SRAUTOSEN.

The trigger condition must be set to SLOW.

**Group**

Search and Mark

**Syntax**

```
<Qstring>
```

**Related Commands**

```
SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:CONDITION
```

**Arguments**

Search<x> is the number of the search.

<Qstring> is the slow channel data binary value.

**Examples**

```
```

sets the slow data search value to "don't care."

```
```

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer 2-875

This command sets or queries the binary slow identifier value to be used when searching on a SENT bus signal.

**Conditions**
Requires option SRAUTOSEN.
The trigger condition must be set to SLOW.

**Group**
Search and Mark

**Syntax**

**Related Commands**
SEARCH:SEARCH<x>:TRIGGER:A:BUS:SENT:CONDITION

**Arguments**
Search<x> is the number of the search.
<Qstring> is the slow channel identifier binary value.

**Examples**
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:IDENTIFIER:VALUE "XXXX" sets the search identifier value to "don't care."

SEARCH:SEARCH<x>:TRIGGER:A:BUS:SOURce

This command sets or queries the bus source for the bus search to determine where to place a mark. The search number is specified by <x>.

**Group**
Search and Mark

**Syntax**
SEARCH:SEARCH<x>:TRIGGER:A:BUS:SOURce {B0|B1|B2|B3|B4|B5|B6|B7|B8|B9|B10|B11|B12|B13|B14|B15|B16}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:SOURce?

**Arguments**
B0–B16 specifies the bus source as a bus number from B01 to B16.
Commands listed in alphabetical order

Examples


SEARCH:SEARCH<x>:TRIGGER:A:BUS:SPI:CONDITION

This command sets or queries the search condition for an SPI bus search to determine where to place a mark. The search number is specified by <x>.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGGER:A:BUS:SPI:CONDITION {DATA|SS|STARTofframe}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:SPI:CONDITION?

Arguments

DATA specifies the trigger condition as Data.

SS specifies the trigger condition as Slave Selection.

STARTofframe specifies the trigger condition as start of frame.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:CONDITION SS sets the trigger condition for SPI bus trigger search 1 to SS.


This command sets or queries the length of the data string in bytes used for the specified SPI bus trigger search to determine where to place a mark. The search condition must be DATA. The search number is specified by <x>.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.
Commands listed in alphabetical order

Group  Search and Mark

Syntax  
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:SIZE  <NR1>  
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:SIZE?

Arguments  <NR1> specifies the number of contiguous data bytes.

Examples  SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:DATA:SIZE 1 sets the length of the data string for SPI bus trigger search 1 to 1 byte.  

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:VALue

The command sets or queries the binary data string used for an SPI bus search to determine where to place a mark. The search number is specified by <x>. The search condition must be DATA.

Conditions  Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group  Search and Mark

Syntax  
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:VALue  <QString>  
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:VALue?

Arguments  <QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format, and A-F, 0-9, and X for hexadecimal format.

Examples  SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:DATA:VALUE "00001111" sets the data value for SPI bus trigger search 1 to "00001111" in binary format.  
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDITION

This command sets or queries the search condition for an SPMI bus.

Conditions
Requires option SRPM

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDITION {SSC|RESet|SLEep|SHUTdown|WAKEup|MASTERREAD|MASTERWRITE|REGREAD|REGWRITE|DEVICEDESCMASTERREAD|DEVICEDESCSLAVEREAD|EXTREGREAD|EXTREGWRITE|LONGEXTREGREAD|LONGEXTREGWRITE|REGWRITE|AUTHenticate|TRANSferbusownership|PARityerror}

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDITION?

Arguments
Search<x> is the search identifier number.

SSC specifies searching for the Sequence Start Condition.

RESet specifies searching for the Reset command sequence.

SLEep specifies searching for the Sleep command sequence.

SHUTdown specifies searching for the Shutdown command sequence.

WAKEup specifies searching for the Wakeup command sequence.

MASTERREAD specifies searching for the Master Read command sequence.

MASTERWRITE specifies searching for the Master Write command sequence.

REGREAD specifies searching for the Register Read command sequence.

REGWRITE specifies searching for the Register Write command sequence.

DEVICEDESCMASTERREAD specifies searching for the Device Descriptor Block Master Read command sequence

DEVICEDESCSLAVEREAD specifies searching for the Device Descriptor Block Slave Read command sequence.

EXTREGREAD specifies triggering on the Extended Register Read command sequence.

EXTREGWRITE specifies searching for the Extended Register Write command sequence.

LONGEXTREGREAD specifies searching for the Extended Register Read Long command sequence.
LONGEXTREGWRITE specifies searching for the Extended Register Write Long command sequence.
REG0WRITE specifies searching for the Register 0 Write command sequence.
AUTHENTICATE specifies searching for the Authentication command sequence.
TRANSFERBUSOWNERSHIP specifies searching for the Transfer Bus Ownership (TBO) command sequence.
PARITYERROR specifies searching for the parity errors.

Examples


This command sets or queries the length of the data string, in bytes, to be used when searching on an SPMI bus signal.

Conditions
Requires option SRPM.
The Search condition must be set to EXTREGREAD, EXTREGWRITE, LONGEXTREGREAD, or LONGEXTREGWRITE.
Note that this a floating point value. If the condition is set to REG0WRITE, the query return value will be a fractional value (0.875) because the data size is less than a byte.

Group
Trigger

Syntax

Related Commands
SEARCH.SEARCH<x>:TRIGGER:A:BUS:SPMI:CONDITION
SEARCH.SEARCH<x>:TRIGGER:A:BUS:SPMI:DATA:VALUE

Arguments
Search<x> is the Search identifier number.
<NR2> is the size of the data string in bytes.
Examples


SEARCH:SEARCH2:TRIGGER:A:BUS:SPMI:DATA:SIZE 4 to indicate the
search data pattern has four bytes of data.


This command specifies the binary data string used for SPMI when
the search condition is MASTERREAD, MASTERWRITE, REGREAD,
REGWRITE, EXTREGREAD, EXTREGWRITE, LONGEXTREGREAD,
LONGEXTREGWRITE, or REG0WRITE.

Conditions
Requires option SRPM

Group
Search and Mark

Syntax

Related Commands
SEARCH:SEARCH<x>:TRIGGER:A:BUS:SPMI:CONDITION

Arguments
Search<x> is the search identifier number.
<Qstring> is the binary data string that identifies the data value for which
to search.

Examples
the binary data value to "don't care" for Search 2.

indicate Search 2 is searching for binary data value XXXX1111.


This command sets or queries the binary data string that identifies the
master address used in SPMI when the search condition is MASTERREAD,
MASTERWRITE, or DEVICEDESCMASTERREAD.

Conditions
Requires option SRPM
Commands listed in alphabetical order

Group Search and Mark

Syntax


Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition

Arguments

Search<x> is the search identifier number.

<Qstring> is the binary data string that identifies the master address for which to search.

Examples

SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:MASTERADDRESS:VALUE "XX" sets the binary master address on which to search to "don't care."


SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:NORESPONSE

This command sets or queries whether or not to search for No Response frames.

Conditions Requires option SRPM

Group Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:NORESPONSE <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:NORESPONSE?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition

Arguments

Search<x> is the search identifier number.

<Qstring> is either "TRUE" or "FALSE" boolean value for searching on SPMI No Response frames.

Examples

Commands listed in alphabetical order

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:NORESPONSE?


This command sets or queries the binary data string that identifies the register address used in SPMI triggering if the trigger condition is MASTERREAD, MASTERWRITE, REGREAD, REGWRITE, EXTREGREAD, EXTREGWRITE, LONGEXTREGREAD, or LONGEXTREGWRITE.

- **Conditions**: Requires option SRPM
- **Group**: Search and Mark
- **Syntax**

  SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:REGISTERADDRESS:VALUE <Qstring>


- **Arguments**

  Search<x> is the search identifier number.

  <Qstring> is the binary data string that identifies the register address for which to search.

- **Examples**

  SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:REGISTERADDRESS:VALUE "XXXXXXXX" sets the binary register address to "don't care."


  might return

  SEARCH:SEARCH5:TRIGger:A:BUS:SPMI:REGISTERADDRESS:VALUE "XXXX1111" to indicate Search 5 is searching for binary register address is XXXX1111.


This command sets or queries the binary data string that identifies the slave address used in SPMI when the search condition is RESET, SLEEP, SHUTDOWN, WAKEUP, AUTHENTicate, REGREAD, REGWRITE, EXTREGREAD, EXTREGWRITE, LONGEXTREGREAD, or DEVICEDESC SLAVE READ.

- **Arguments**

  Search<x> is the search identifier number.

  <Qstring> is the binary data string that identifies the slave address for which to search.

- **Examples**

  SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:SLAVEADDRESS:VALUE "XXXXXXXX" sets the binary slave address to "don't care."


  might return

  SEARCH:SEARCH5:TRIGger:A:BUS:SPMI:SLAVEADDRESS:VALUE "XXXX1111" to indicate Search 5 is searching for binary slave address is XXXX1111.
Commands listed in alphabetical order

**Conditions**
Requires option SRPM

**Group**
Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:SLAVEADDRESS:VALUE <Qstring>
```
```
```

**Related Commands**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition
```

**Arguments**
Search<x> is the search identifier number.

<Qstring> is the binary data string that identifies the slave address for which to search.

**Examples**
```
sets the binary slave address to 0011 on Search 3.
```
```
SEARCH:SEARCH1:TRIGger:A:BUS:SPMI:SLAVEADDRESS:VALUE "1010" to indicate the binary slave address is 1010 for Search 1.
```


This command sets or queries the high binary address value used when searching on a USB bus signal to determine where to place a mark. The search number is specified by <x>. The search condition must be set to TOKEN.

The VALue and HIVALue set a range that the INrange and OUTrange qualifiers use to decide when to trigger. For example, if the QUALIFER is set to INrange, and the address is within the range set by VALue and HIVALue, then a trigger can be generated.

**Conditions**
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
```
```
```
```
## SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDdress:VALue

This command sets or queries the binary address value used for a USB bus search to determine where to place a mark. The search number is specified by `<x>`. The search condition must be set to TOKEN.

### Conditions
Requires option 5-SRUSB 2 or SUP5-SRUSB 2 Triggering and Analysis application.

### Group
Search and Mark

### Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDdress:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDdress:VALue?
```

### Related Commands

### Arguments
<QString> specifies the data value in the specified valid format. Valid characters are 0, 1, and X for binary; A-F, 0-9 and X for hexadecimal; and for symbolic.

### Examples
```
indicating that the address value for normal token for USB bus trigger search 1 is
set to "XX," in hexadecimal format.
```

```
```

### Related Commands

### Arguments
<QString> specifies the data value in the specified valid format. Valid characters are 0, 1, and X for binary; A-F, 0-9 and X for hexadecimal; and for symbolic.

### Examples
```
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ADDRESS:VALUE "1110000"
sets the address value for the normal token for USB bus trigger search 1 to "1110000" in binary format.

indicating that the address value for normal token for USB bus trigger search 1 is set to
"11" hexadecimal format.
```
SEARCH:SEARCH<x>:TRIGGER:A:BUS:USB:CONDITION

This command sets or queries the search condition for a USB bus search to determine where to place a mark. The search number is specified by <x>.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGGER:A:BUS:USB:CONDITION

{DATAPacket|EOP|ERROR|HANDSHAKEPacket|RESET|RESUME|
SPECIALPacket|SUSPEND|SYNC|TOKENPacket}

SEARCH:SEARCH<x>:TRIGGER:A:BUS:USB:CONDITION?

Arguments

DATAPacket specifies the search condition as Data Packet.

EOP specifies the search condition as End of Packet.

ERROR specifies the search condition as Error.

HANDSHAKEPacket specifies the search condition as Handshake Packet.

RESET specifies the search condition as Reset.

RESUME specifies the search condition as Resume.

SPECIALPacket specifies the search condition as Special Packet.

SUSPEND specifies the search condition as Suspend.

SYNC specifies the search condition as Sync.

TOKENPacket specifies the search condition as Token (Address) Packet.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:CONDITION EOP sets the search condition for USB bus trigger search 1 to End of Packet.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:CONDITION SYNC, indicating that the search condition for USB bus trigger search 1 is set to Sync.


This command sets or queries the high binary data value used with In Range and Out of Range qualifiers for a USB bus search to determine where to place a mark. The search number is specified by <x>. The search condition must be set to DATA.

This command sets the byte offset to look for a data pattern at, in bytes, to be used when searching on a USB bus signal. The search number is specified by `<x>`. The search condition must be set to DATA.

**Conditions**
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
SEARCH:SEARCH<x>:TRIGGER:A:BUS:USB:DATA:OFFSET <NR1>

**Arguments**
<NR1> specifies the data offset in bytes.

**Examples**
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:OFFSET 5 sets the data offset for USB bus trigger search 1 to 5 bytes.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:QUALifier**

This command sets or queries the qualifier to be used when searching on a USB bus signal. The search condition must be set to IDANDDATA OR DATA. The search number is specified by `<x>`.

**Conditions**
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:QUALifier {EQUAL|UNEQUAL|
LESSThan|MOREThan|LESSEQqual|MORREEQual|INrange|OUTrange}
```

**Arguments**
- `EQUAL` specifies the qualifier as Equal.
- `INrange` specifies the qualifier as Inside Range.
- `LESSEQqual` specifies the qualifier as Less Than or Equal to.
- `MORREEQual` specifies the qualifier as More Than or Equal to.
- `OUTrange` specifies the qualifier as Out of Range.
- `UNEQUAL` specifies the qualifier as Unequal.
- `LESSThan` specifies the qualifier as Less Than.
- `MOREThan` specifies the qualifier as More Than.

**Examples**
```
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:QUALIFIER INRANGE sets the qualifier to INRANGE.
```

**SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:SIZe**

This command sets or queries the length of the data string, in bytes, used for a USB bus search to determine where to place a mark. The search number is specified by `<x>`. The search condition must be set to DATA.

**Conditions**
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.
Commands listed in alphabetical order

**Group**  
Search and Mark

**Syntax**  
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:SIZE <NR1>  

**Arguments**  
<NR1> specifies the data size in bytes.

**Examples**  
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:SIZE 1 sets the size of the data string for USB bus trigger search 1 to 1 byte.  

**SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:TYPe**

This command sets or queries the USB bus search type. The search number is specified by <x>.

**Conditions**  
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

**Group**  
Search and Mark

**Syntax**  
{ANY|DATA0|DATA1|DATA2|MDATA}  
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:TYPe?

**Arguments**  
ANY specifies the data packet type as Any.  
DATA0 specifies the data packet type as DATA0.  
DATA1 specifies the data packet type as DATA1.  
DATA2 specifies the data packet type as DATA2.  
MDATA specifies the data packet type as MDATA.

**Examples**  
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:TYPe ANY sets the data packet type for USB bus trigger search 1 to ANY.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:VALue

This command sets or queries the binary data value used for a USB bus search to determine where to place a mark. The search number is specified by <x>. The search condition must be set to DATA.

**Arguments**

<QString> specifies the data value. The valid characters are 0, 1, and X for binary format, and A-F, 0-9, and X for hexadecimal format.

**Examples**

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:VALUE "AB" sets the data value for data token for USB bus trigger search 1 to "AB" in hexadecimal format.


SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ENDPoint:VALue

This command sets or queries the endpoint binary value used for a USB bus search to determine where to place a mark. The search number is specified by <x>. The search condition must be set to TOKEN.

**Conditions**

Requires option 5-SRUSB 2 or SUP5-SRUSB 2 Triggering and Analysis application.

**Group**

Search and Mark
Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ENDPoint:VALUE <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ENDPoint:VALUE?

Arguments
QString specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format; A-F, 0-9 and X for hexadecimal format.

Examples
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ENDPOINT:VALUE "1111" sets the endpoint value for normal token for USB bus trigger search 1 to "1111" in binary format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ERRTYPE

This command sets or queries the error type for a USB bus search to determine where to place a mark. The search number is specified by <x>.

Conditions
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ERRTYPE {BITSTUFFing|CRC5|CRC16|PID}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ERRTYPE?

Arguments
BITSTUFFing specifies the error type as Bit Stuffing.
CRC5 specifies the error type as Token CRC5 (Cyclic Redundancy Check 5).
CRC16 specifies the error type as Data CRC16 (Cyclic Redundancy Check 16).
PID specifies the error type as PID Check Bits.

Examples
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ERRTYPE CRC5 sets the error type for USB bus trigger search 1 to Token CRC5.
**SEARCH:SEARCH<\text{x}>:TRIGger:A:BUS:USB:HANDSHAKETYPE**

This command sets or queries the handshake type for the specified USB bus trigger search to determine where to place a mark. The search number is specified by <\text{x}>.

**Conditions**
Requires option 5-SRUSB or SUP5-SRUSB Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
```
SEARCH:SEARCH<\text{x}>:TRIGger:A:BUS:USB:HANDSHAKETYPE {ACK|ANY|NAK|NYET|STALL}
SEARCH:SEARCH<\text{x}>:TRIGger:A:BUS:USB:HANDSHAKETYPE?
```

**Arguments**
- **ACK** specifies the handshake type as Acknowledgement (positive) (XX10).
- **ANY** specifies the handshake type as Any (0010).
- **NAK** specifies the handshake type as Negative Acknowledgment (1010).
- **NYET** specifies the handshake type as No response Yet (0110).
- **STALL** specifies the handshake type as Stall (endpoint is halted or control pipe request not supported) (1110).

**Examples**
```
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:HANDSHAKETYPE NAK sets the handshake type for USB bus trigger search 1 to NAK.
```

**SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SOFFRAMENUMber**

This command sets or queries the frame number string to use for the Start of Frame for the specified USB bus trigger search to determine where to place a mark. The search number is specified by <\text{x}>. The search condition must be set to TOKEN.

**Conditions**
Requires option 5-SRUSB or SUP5-SRUSB Triggering and Analysis application.

**Group**
Search and Mark
Commands listed in alphabetical order

**Syntax**
SEARCH:SEARCH<x>:TRIGGER:A:BUS:USB:SOFRAMENUMBER <QString>
SEARCH:SEARCH<x>:TRIGGER:A:BUS:USB:SOFRAMENUMBER?

**Arguments**
<QString> specifies the frame number string for the Start of Frame in a valid format (binary, hexadecimal, or symbolic).

**Examples**
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SOFRAMENUMBER "1001" sets the frame number string for the SOF for USB bus trigger search 1 to "1001" in symbolic format.


---

**SEARCH:SEARCH<x>:TRIGGER:A:BUS:USB:SPECIALType**

This command sets or queries the USB search type for the specified USB bus trigger search to determine where to place a mark. The search number is specified by <x>.

**Conditions**
Requires option 5-SRUSB 2 or SUP5-SRUSB 2 Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
SEARCH:SEARCH<x>:TRIGGER:A:BUS:USB:SPECIALType
{ANY|ERROR|PING|PREamble|RESERVED|SPLIT}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:USB:SPECIALType?

**Arguments**
ANY specifies the PID value as Any (XX00).
ERROR specifies the PID value as ERR (1100).
PING specifies the PID value as PING (0100).
PREamble specifies the PID value as PRE (1100).
RESERVED specifies the PID value as Reserved (0000).
SPLIT specifies the PID value as Split (1000).

**Examples**
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPECIALTYPE ERROR sets the PID value for USB bus trigger search 1 to ERROR.
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPECIALTYPE SPLIT, indicating
that the PID value for USB bus trigger search 1 is set to SPLIT.


This command sets or queries the Endpoint Type value for the specified USB bus trigger search on split token field to determine where to place a mark. The search number is specified by <x>.

Conditions
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group
Search and Mark

Syntax
{BULK|CONTROL|NOCARE|INTERRUPT|ISOchronous}

Arguments
BULK specifies the Endpoint Type value as Bulk (10).
CONTROL specifies the Endpoint Type value as Control (00).
NOCARE specifies the Endpoint Type value as Don’t Care.
INTERRUPT specifies the Endpoint Type value as Interrupt (11).
ISOchronous specifies the Endpoint Type value as Isochronous (01).

Examples
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:ET:VALUE DONTCARE, indicating that the Endpoint Type value for USB bus trigger search 1 is set to DONTCare.


This command sets or queries the binary hub address value to be used when searching on a USB bus signal. The search number is specified by <x>. The search condition must be set to Special with packet type SPLIT.

This command sets or queries the binary port address used when searching on a USB bus signal. The search number is specified by `<x>`. The search condition must be set to Special with a packet type SPLIT.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Search and Mark</td>
</tr>
<tr>
<td>Arguments</td>
<td><code>&lt;QString&gt;</code> specifies the hub address. The valid characters are .</td>
</tr>
<tr>
<td>Examples</td>
<td><code>SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:HUB:VALUE &quot;1001&quot;</code> sets the hub address to &quot;XXX1001&quot;.</td>
</tr>
</tbody>
</table>


This command sets or queries the binary port address used when searching on a USB bus signal. The search number is specified by `<x>`. The search condition must be set to Special with a packet type SPLIT.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Search and Mark</td>
</tr>
<tr>
<td>Arguments</td>
<td><code>&lt;QString&gt;</code> specifies the port address in the valid format. The valid characters are</td>
</tr>
<tr>
<td>Examples</td>
<td><code>SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:PORT:VALUE &quot;1111&quot;</code> sets the port address for USB bus trigger search 1 to &quot;XXX1111&quot; for binary format.</td>
</tr>
</tbody>
</table>
indicating that the port address for USB hub trigger search 1 is set to "XX" in hexadecimal format.


This command sets or queries the Start/Complete value for the specified USB bus trigger on split token field search to determine where to place a mark. The search number is specified by <x>.

**Conditions**
Requires option 5-SRUSB 2 or SUP5-SRUSB 2 Triggering and Analysis application.

**Group**
Search and Mark

**Syntax**
{CSPLIT|NOCARE|SSPLIT}

**Arguments**
CSPLIT specifies Complete (CSPLIT) (1) Start/Complete value.
NOCARE specifies the Start/Complete value as Don’t Care (X).
SSPLIT specifies Start (SSPLIT) (0) Start/Complete value.

**Examples**


This command sets or queries the Start/End value for the specified USB bus trigger on split token field search to determine where to place a mark. The search number is specified by <x>.

**Conditions**
Requires option 5-SRUSB 2 or SUP5-SRUSB 2 Triggering and Analysis application.

**Group**
Search and Mark
Syntax
{NOCARE|FULLSPEED|ISOALL|ISOEND|ISOMID|ISOSTART|LOWSPEED}

Arguments
NOCARE specifies Don’t Care (X) Start/End bit value.
FULLSPEED specifies Control/Bulk/Interrupt Full Speed device (0X) Start/End bit value.
ISOALL specifies Isochronous Data is All (11) Start/End bit value.
ISOEND specifies Isochronous Data is End (01) Start/End bit value.
ISOMID specifies Isochronous Data is Middle (00) Start/End bit value.
ISOSTART specifies Isochronous Data is Start (10) Start/End bit value.
LOWSPEED specifies Control/Bulk/Interrupt Low Speed device (1X) Start/End bit value.

Examples
SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:SE:VALUE ISOSTART sets the Start/End value for USB bus trigger on split token field search 1 to ISOSTART.

SEARCH:SEARCH<x>:TRIGGER:A:BUS:USB:TOKENType

This command sets or queries the token type used to search a USB bus signal. The search number is specified by <x>.

Conditions
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGGER:A:BUS:USB:TOKENType
{ANY|IN|OUT|SETUP|SOF}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:USB:TOKENType?

Arguments
ANY specifies the token type as Any (XX01).
IN specifies the token type as IN (1001).
OUT specifies the token type as OUT (0001).
SETUP specifies the token type as SETUP (1101).
SOF specifies the token type as Start Of Frame (0101).

**Examples**
SEARCH: SEARCH1:TRIGGER:A:BUS:USB:TOKENTYPE IN sets the token type for USB bus trigger search 1 to IN.
SEARCH: SEARCH1:TRIGGER:A:BUS:USB:TOKENTYPE SOF, indicating that the
token type for USB bus trigger search 1 is set to Start Of Frame.

**SEARCH: SEARCH<x>:TRIGGER:A:DDRREAD:DATARate**

This command sets or queries the DDR read search data rate for DDR3 and LPDDR3 standards of the specified search.

**Conditions**
Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Set the DDR Read standard (DDR3 or LPDDR3) before using this command.
See Related Commands.

**Syntax**
SEARCH: SEARCH<x>:TRIGGER:A:DDRREAD:DATARate {333|800|1066|1200|1333|1466|1600|1866|2133}
SEARCH: SEARCH<x>:TRIGGER:A:DDRREAD:DATARate?

**Related Commands**
SEARCH: SEARCH<x>:TRIGGER:A:DDRREAD:STANDARD

**Arguments**
333–2133 sets the data rate to the specified value.
The DDR3 standard supports the following data rates:
800|1066|1333|1600|1866|2133.
The LPDDR3 standard supports the following data rates:
333|800|1066|1200|1333|1466|1600|1866|2133.

**Examples**
SEARCH: SEARCH2:TRIGGER:A:DDRREAD:DATARate 1333 sets the data rate to 1333 for DDR read Search 2.
SEARCH: SEARCH1:TRIGGER:A:DDRREAD:DATARate? might return
SEARCH: SEARCH1:TRIGGER:A:DDRREAD:DATARate 800 indicating that the data rate for DDR read Search 1 is 800.
SEARCH:SEARCH<x>:TRIGGER:A:DDRREAD:DATASource

This command sets or queries the DDR read data source when the search type is DDR READ.

Conditions
Requires option 6-DBDDR3
Requires 6 Series MSO oscilloscope
Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group
Search

Syntax
SEARCH:SEARCH<x>:TRIGGER:A:DDRREAD:DATASource
{CH<x>|CH<x>_D<x>|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGGER:A:DDRREAD:DATASource?

Arguments
CH<x> specifies channel <x> as the DDR read data source for the specified search <x>.
CH<x>_D<x> specifies digital waveform <x> of channel <x> as the DDR read data source for the specified search <x>.
Math<x> specifies math waveform <x> as the DDR read data source for the specified search <x>.
REF<x> specifies reference waveform <x> as the DDR read data source for the specified search <x>.
REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR read data source for the specified search <x>.

Examples

SEARCH:SEARCH<x>:TRIGGER:A:DDRREAD:HYSteresis

This command sets or queries the DDR read hysteresis reference level value, when the search type is DDR READ.

Conditions
Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group Search

Syntax
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:HYSteresis <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:HYSteresis?

Related Commands
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

Arguments
NR3 sets the DDR read search hysteresis percent value in the range of 0% to 50%.

Examples
SEARCH:SEARCH2:TRIGger:A:DDRREAD:HYSteresis 15 sets the DDR read search hysteresis value to 15%.
SEARCH:SEARCH1:TRIGger:A:DDRREAD:HYSteresis? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:HYSteresis 10, indicating that the DDR read search hysteresis value is set to 10%.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MARgin

This command sets or queries the DDR read margin reference level value, when the search type is DDR READ.

Conditions
Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group Search

Syntax
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MARgin <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MARgin?

Related Commands
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

Arguments
NR3 sets the DDR read search margin percent value in the range of 0% to 100%.
Commands listed in alphabetical order

**Examples**

SEARCH:SEARCH2:TRIGGER:A:DDRREAD:MARGIN 12 sets the DDR read search margin value to 12%.

SEARCH:SEARCH1:TRIGGER:A:DDRREAD:MARGIN 10, indicating that the DDR read search margin value is set to 10%.

**SEARCH:SEARCH<x>:TRIGGER:A:DDRREAD:POSTAMBLE:LENGTH**

This command sets or queries the DDR read postamble length when the search type is DDR READ.

**Conditions**

Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.

**Group**

Search

**Syntax**

SEARCH:SEARCH<x>:TRIGGER:A:DDRREAD:POSTAMBLE:LENGTH
{500E-3|500E-1.5|500E-1}

SEARCH:SEARCH<x>:TRIGGER:A:DDRREAD:POSTAMBLE:LENGTH?

**Arguments**

500E-1.5 sets the postamble length to 1.5 tCK (number of clock cycles).
500E-1 sets the postamble length to 1 tCK (number of clock cycles).
500E-3 sets the postamble length to 0.5 tCK (number of clock cycles).

**Examples**

SEARCH:SEARCH2:TRIGGER:A:DDRREAD:POSTAMBLE:LENGTH 500E-3 sets the DDR read postamble length for Search 2 to 0.5 tCK.

SEARCH:SEARCH1:TRIGGER:A:DDRREAD:POSTAMBLE:LENGTH 500E-1, indicating that the read postamble length for Search 1 is set to 1 tCK.

**SEARCH:SEARCH<x>:TRIGGER:A:DDRREAD:PREAMBLE:TYPE**

This command sets or queries the DDR read preamble type when the search type is DDR READ.

**Conditions**

Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Group  Search

Syntax  SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:PREAMBLE:TYPE {STATIC|DYNAMIC}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:PREAMBLE:TYPE?

Arguments  STATIC sets the DDR Read preamble type to Static.
DYNAMIC sets the DDR Read preamble type to Dynamic.


This command sets or queries the DDR read data high reference level value, when the search type is DDR READ.

Conditions  Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Set the DDR read reference level mode to Manual before using this command.
See Related Commands.

Group  Search and Mark


Related Commands  SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevelMode

Arguments  NR3 sets the DDR read search data high reference value.

might return SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH 50E-3, indicating that the DDR read search data high reference value is set to 50E-3.


This command sets or queries the DDR read data low reference level value, when the search type is DDR READ.

Conditions
Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW?

Related Commands
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

Arguments
NR3 sets the DDR read search data low reference value.

Examples


This command sets or queries the DDR read data mid reference level value, when the search type is DDR READ.

Conditions
Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Commands listed in alphabetical order

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID?

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

Arguments NR3 sets the DDR read search data mid reference value.


This command sets or queries the DDR read reference level strobe high value, when the search type is DDR READ.

Conditions Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark


Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

Arguments NR3 sets the DDR read search strobe high value.
Examples


This command sets or queries the DDR read reference level strobe low value, when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO oscilloscope.

Set the DDR read reference level mode to Manual before using this command.

See Related Commands.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLeVel:Mode

Arguments

NR3 sets the DDR read search strobe low reference value.

Examples


This command sets or queries the DDR read reference level strobe mid value, when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFlevel:STROBE:MID?
```

Related Commands

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode
```

Arguments

NR3 sets the DDR read search strobe mid reference value.

Examples

```
sets the DDR read search strobe mid reference value.

50E-3, indicating that the DDR read search strobe mid reference value is set
to 50E-3.
```

**SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode**

This command sets or queries the DDR read reference level mode to auto or
manual, when the search type is DDR READ.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO oscilloscope.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode
{AUTO|MANUAL}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode?
```

Related Commands

```
```
## Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO</td>
<td>sets the DDR read reference level mode to auto.</td>
</tr>
<tr>
<td>MANUAL</td>
<td>sets the DDR read reference level mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.</td>
</tr>
</tbody>
</table>

## Examples


### SEARCH:SEARCH\<x\>:TRIGger:A:DDRREAD:STANdard

This command sets or queries the DDR read search standard as DDR3 or LPDDR3.

#### Conditions

Requires option 6-DBDDR3
Requires 6 Series MSO oscilloscope

#### Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STANDard {DDR3|LPDDR3}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STANDard?
```

#### Arguments

- **DDR3**: sets the DDR read search standard as DDR3.
- **LPDDR3**: sets the DDR read search standard as LPDDR3.

#### Examples


### SEARCH:SEARCH\<x\>:TRIGger:A:DDRREAD:STROBESource

This command sets or queries the DDR read strobe source when the search type is DDR READ.

#### Conditions

Requires option 6-DBDDR3
Requires 6 Series MSO oscilloscope

Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STROBESource
{CH<x>|CH<x>_D<x>|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STROBESource?

Arguments
CH<x> specifies channel <x> as the DDR read strobe trigger source for the specified search <x>.

CH<x>_D<x> specifies digital waveform <x> of channel <x> as the DDR read strobe trigger source for the specified search <x>.

Math<x> specifies math waveform <x> as the DDR read strobe trigger source for the specified search <x>.

REF<x> specifies reference waveform <x> as the DDR read strobe trigger source for the specified search <x>.

REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR read strobe trigger source for the specified search <x>.

Examples


SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATARate

This command sets or queries the DDR read/write data rate for DDR3 and LPDDR3 standards of the specified search.

Conditions
Requires option 6-DBDDR3.

Requires a 6 Series MSO oscilloscope.

Set the DDR Read/Write standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group
Search and Mark
Syntax
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATARate
{333|800|1066|1200|1333|1466|1600|1866|2133}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATARate?

Related Commands
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard

Arguments
333–2133 sets the data rate to the specified value.

The DDR3 standard supports the following data rates:
800|1066|1333|1600|1866|2133.

The LPDDR3 standard supports the following data rates:
333|800|1066|1200|1333|1466|1600|1866|2133.

Examples
SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:DATARate 1333 sets the data rate to 1333 for DDR read Search 2.
SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:DATARate? might return SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:DATARate 1466 indicating that the data rate for DDR read Search 5 is 1466.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATASource

This command sets or queries the DDR read/write data source when the search type is DDR READWRITE.

Conditions
Requires option 6-DBDDR3
Requires 6 Series MSO oscilloscope
Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATASource
{CH<x>|CH<x>_D<x>|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATASource?

Arguments
CH<x> specifies channel <x> as the DDR read/write data source for the specified search <x>.
CH<x>_D<x> specifies digital waveform <x> of channel <x> as the DDR read/write data source for the specified search <x>. 
Math<x> specifies math waveform <x> as the DDR read/write data source for the specified search <x>.

REF<x> specifies reference waveform <x> as the DDR read/write data source for the specified search <x>.

REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR read/write data source for the specified search <x>.

**Examples**

SEARCH:SEARCH3:TRIGGER:A:DDRREADWRITE:DATASource CH2 sets the source waveform of Search 3 to Channel 2 for the DDR read/write search.


**SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:HYSteresis**

This command sets or queries the DDR read/write hysteresis reference level value, when the search type is DDR READWRITE.

**Conditions**

Requires option 6-DBDDR3.

Requires a 6 Series MSO oscilloscope.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

**Group**

Search and Mark

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:HYSteresis <NR3>

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:HYSteresis?

**Related Commands**

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

**Arguments**

NR3 sets the DDR read/write search hysteresis percent value in the range of 0% to 50%.

**Examples**

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:HYSteresis 10 sets the DDR read/write search hysteresis value to 10%.

SEARCH:SEARCH1:TRIGGER:A:DDRREADWRITE:HYSteresis? might return SEARCH:SEARCH1:TRIGGER:A:DDRREADWRITE:HYSteresis 5, indicating that the DDR read/write search hysteresis value is set to 5%.
**SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MARgin**

This command sets or queries the DDR read/write margin reference level value, when the search type is DDR READWRITE.

**Conditions**
- Requires option 6-DBDDR3.
- Requires a 6 Series MSO oscilloscope.
- Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

**Group**
Search and Mark

**Syntax**
```plaintext
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MARgin <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MARgin?
```

**Related Commands**
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVEL:Mode

**Arguments**
- `NR3` sets the DDR read/write search margin percent value in the range of 0% to 100%.

**Examples**
- `SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:MARgin 15` sets the DDR read/write search margin value to 15%.
- `SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:MARgin?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:MARgin 10`, indicating that the DDR read/write search margin value is set to 10%.

---

**SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth**

This command sets or queries the DDR read/write postamble length when the search type is DDR READWRITE.

**Conditions**
- Requires option 6-DBDDR3.
- Requires a 6 Series MSO oscilloscope.

**Group**
Search and Mark

**Syntax**
```plaintext
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth {500E-3|1.5|1}
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGTH?

**Arguments**
- 500E-1.5 sets the postamble length to 1.5 tCK (number of clock cycles).
- 500E-1 sets the postamble length to 1 tCK (number of clock cycles).
- 500E-3 sets the postamble length to 0.5 tCK (number of clock cycles).

**Examples**
  500E-1.5 sets the DDR read/write postamble length for Search 3 to 1.5 tCK.
  500E-1, indicating that the read/write postamble length for Search 2 is set to 1 tCK.

**SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE**

This command sets or queries the DDR read/write preamble type when the search type is DDR READWRITE.

**Conditions**
- Requires option 6-DBDDR3.
- Requires a 6 Series MSO oscilloscope.

**Group**
- Search and Mark

**Syntax**
- SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE {STATIC|DYNAMIC}
- SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE?

**Arguments**
- STATIC sets the DDR Read/Write preamble type to Static.
- DYNAMIC sets the DDR Read/Write preamble type to Dynamic.

**Examples**
  sets the search 2 DDR read preamble type to Static.
  DYNAMIC, indicating that the DDR read/write preamble type for Search 5 is set to dynamic mode.

This command sets or queries the DDR read/write data high reference level value, when the search type is DDR READWRITE.

**Conditions**

Requires option 6-DBDDR3.

Requires a 6 Series MSO oscilloscope.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

**Group**

Search and Mark

**Syntax**


<NR3>

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH?

**Related Commands**

SEARCH:SEARCH<x>:TRIGger:A DDRREADWRITE REFLEVELMode

**Arguments**

NR3 sets the DDR read/write search data high reference value.

**Examples**


3.8E-3 sets the DDR read/write search data high reference value.


might return


50E-3, indicating that the DDR read/write search data high reference value is set to 50E-3.

---


This command sets or queries the DDR read/write data low reference level value, when the search type is DDR READWRITE.

**Conditions**

Requires option 6-DBDDR3.

Requires a 6 Series MSO oscilloscope.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.
Group | Search and Mark

Syntax

<NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW?

Related Commands | SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

Arguments | NR3 sets the DDR read/write search data low reference value.

Examples

3.8E-3 sets the DDR read/write search data low reference value.

SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW?
50E-3, indicating that the DDR read/write search data low reference value is
set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID

This command sets or queries the DDR read/write data mid reference level value,
when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Set the DDR read/write reference level mode to Manual before using this
cmdon. See Related Commands.

Group | Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID
<NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID?

Related Commands | SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

Arguments | NR3 sets the DDR read/write search data mid reference value.
Examples


This command sets or queries the DDR read/write reference level strobe high value, when the search type is DDR READWRITE.

Conditions
Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH?

Related Commands
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

Arguments
NR3 sets the DDR read search strobe high value.

Examples

SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH 50E-3, indicating that the DDR read/write search strobe high value is set to 50E-3.


This command sets or queries the DDR read/write reference level strobe low value, when the search type is DDR READWRITE.
Conditions
Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW?

Related Commands
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

Arguments
NR3 sets the DDR read/write search strobe low reference value.

Examples
SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW 50E-3, indicating that the DDR read/write search strobe low reference value is set to 50E-3.

This command sets or queries the DDR read/write reference level strobe mid value, when the search type is DDR READWRITE.

Conditions
Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID?
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

This command sets or queries the DDR read/write reference level mode to auto or manual, when the search type is DDR READWRITE.

**Arguments**
- AUTO sets the DDR read/write reference level mode to auto.
- MANUAL sets the DDR read/write reference level mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.

**Examples**

**SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard**

This command sets or queries the DDR read/write search standard as DDR3 or LPDDR3.

**Conditions**
- Requires option 6-DBDDR3.
- Requires a 6 Series MSO oscilloscope.

**Group**
- Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard
{DDR3|LPDDR3}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard?
```

**Arguments**
- DDR3 sets the DDR read/write search standard as DDR3.
- LPDDR3 sets the DDR read/write search standard as LPDDR3.

**Examples**
```

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:STANdard LPDDR3 indicating that the DDR standard is set to LPDDR3 for DDR read/write Search 2.
```

**SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STROBESource**

This command sets or queries the DDR readwrite strobe source when the search type is DDR READWRITE.

**Conditions**
- Requires option 6-DBDDR3.
- Requires a 6 Series MSO oscilloscope.
- Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.
Group  
Search and Mark

Syntax  
SEARCH:SEARCH<x>:TRIGGER:A:DDRWRITE:STROBESource 
{CH<x>|CH<x>_D<x>|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGGER:A:DDRWRITE:STROBESource?

Arguments  
CH<x> specifies channel <x> as the DDR read/write strobe source for the specified search <x>.
CH<x>_D<x> specifies digital waveform <x> of channel <x> as the DDR read/write strobe source for the specified search <x>.
Math<x> specifies math waveform <x> as the DDR read/write strobe source for the specified search <x>.
REF<x> specifies reference waveform <x> as the DDR read/write strobe source for the specified search <x>.
REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR read/write strobe source for the specified search <x>.

Examples  
SEARCH:SEARCH5:TRIGGER:A:DDRWRITE:STROBESource CH8 sets the Search 5 source waveform to Channel 8 for the DDR read/write strobe search.

SEARCH:SEARCH<x>:TRIGGER:A:DDRWRITE:DATARate

This command sets or queries the DDR write search data rate for DDR3 and LPDDR3 standards of the specified search.

Conditions  
Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Set the DDR Write standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group  
Search and Mark

Syntax  
SEARCH:SEARCH<x>:TRIGGER:A:DDRWRITE:DATARate 
{333|800|1066|1200|1333|1466|1600|1866|2133}
SEARCH:SEARCH<x>:TRIGGER:A:DDRWRITE:DATARate?
Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard

Arguments

333–2133 sets the data rate to the specified value.

The DDR3 standard supports the following data rates: 800|1066|1333|1600|1866|2133.

The LPDDR3 standard supports the following data rates: 333|800|1066|1200|1333|1466|1600|1866|2133.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:DATA Rate 1333 sets the data rate to 1333 for DDR read Search 2.

SEARCH:SEARCH7:TRIGger:A:DDRWRITE:DATA Rate? might return SEARCH:SEARCH7:TRIGger:A:DDRWRITE:DATA Rate 1466 indicating that the data rate for DDR read Search 7 is 1466.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATA Source

This command sets or queries the DDR write data source when the search type is DDR Write.

Conditions

Requires option 6-DBDDR3

Requires 6 Series MSO oscilloscope

Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATA Source
{CH<x>|CH<x>_D<x>|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATA Source?

Arguments

CH<x> specifies channel <x> as the DDR write data source for the specified search <x>.

CH<x>_D<x> specifies digital waveform <x> of channel <x> as the DDR write data source for the specified search <x>.

Math<x> specifies math waveform <x> as the DDR write data source for the specified search <x>.

REF<x> specifies reference waveform <x> as the DDR write data source for the specified search <x>.
Commands listed in alphabetical order

REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR write data source for the specified search <x>.

Examples
SEARCH:SEARCH1:TRIGGER:A:WRITE:DATASource CH2 sets the search 1 source waveform to Channel 2 for the DDR write search.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:HYSteresis

This command sets or queries the DDR write hysteresis reference level value, when the search type is DDR WRITE.

Conditions
Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Set the DDR write reference level mode to Manual before using this command.
See Related Commands.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:HYSteresis <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:HYSteresis?

Related Commands
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode

Arguments
NR3 sets the DDR write search hysteresis percent value in the range of 0% to 50%.

Examples
SEARCH:SEARCH2:TRIGger:A:DDRWRITE:HYSteresis 7 sets the DDR write search hysteresis value to 7%.
SEARCH:SEARCH1:TRIGger:A:DDRWRITE:HYSteresis? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:HYSteresis 10, indicating that the DDR write search hysteresis value is set to 10%.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MARgin

This command sets or queries the DDR write margin reference level value, when the search type is DDR WRITE.
Commands listed in alphabetical order

**SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MARgin <NR3>**

Sets the DDR write search margin percent value in the range of 0% to 100%.

**SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MARgin?**

Query the DDR write search margin value.

**Related Commands**

**SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode**

**Examples**

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:MARgin 11 sets the DDR write search margin value to 11%.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:MARgin? might return

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:MARgin 5, indicating that the DDR write search margin value is set to 5%.

**SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth**

This command sets or queries the DDR write postamble length when the search type is DDR WRITE.

**Conditions**

Requires option 6-DBDDR3.

Requires a 6 Series MSO oscilloscope.

**Group**

Search and Mark

**Syntax**

**SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth**

500E-1.5 sets the postamble length to 1.5 tCK (number of clock cycles).

500E-1 sets the postamble length to 1 tCK (number of clock cycles).

500E-3 sets the postamble length to 0.5 tCK (number of clock cycles).

500E-3 sets the postamble length to 0.5 tCK (number of clock cycles).
Examples

SEARCH:SEARCH6:TRIGger:A:DDRWRITE:POSTAMBLE:LENGTH 500E-3 sets the DDR write postamble length for Search 6 to 0.5 tCK.


SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:PREAMBLE:TYPE

This command sets or queries the DDR write preamble type when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO oscilloscope.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:PREAMBLE:TYPE {STATIC|DYNAMIC}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:PREAMBLE:TYPE?

Arguments

STATIC sets the DDR Write preamble type to Static.

DYNAMIC sets the DDR Write preamble type to Dynamic.

Examples


SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH

This command sets or queries the DDR write data high reference level value, when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO oscilloscope.

Set the DDR write reference level mode to Manual before using this command.

See Related Commands.
Commands listed in alphabetical order

**Group**  
Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH?
```

**Related Commands**  
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode

**Arguments**
NR3 sets the DDR read search data high reference value.

**Examples**
```
```
sets the DDR write search data high reference value.

```
50E-3, indicating that the DDR write search data high reference value is set
to 50E-3.
```

**SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW**

This command sets or queries the DDR write data low reference level value,  
when the search type is DDR WRITE.

**Conditions**
Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Set the DDR write reference level mode to Manual before using this command.  
See Related Commands.

**Group**  
Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW?
```

**Related Commands**  
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode

**Arguments**
NR3 sets the DDR read search data low reference value.

**Examples**
```
```
sets the DDR write search data low reference value.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:MID

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH
Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group: Search and Mark

Syntax:

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH
NR3
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH?
```

Related Commands:

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode
```

Arguments:

NR3 sets the DDR write search strobe high value.

Examples:

```
3.8E-3 sets the DDR write search strobe high value.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH?
50E-3, indicating that the DDR write search strobe high value is set to 50E-3.
```

**SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW**

This command sets or queries the DDR write reference level strobe low value, when the search type is DDR WRITE.

Conditions

- Requires option 6-DBDDR3.
- Requires a 6 Series MSO oscilloscope.
- Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group: Search and Mark

Syntax:

```
NR3
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW?
```

Related Commands:

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode
```

Arguments:

NR3 sets the DDR read search strobe low reference value.
Examples

```
```
sets the DDR write search strobe low reference value.

```
50E-3, indicating that the DDR write search strobe low reference value is set
to 50E-3.
```


This command sets or queries the DDR write reference level strobe mid value,
when the search type is DDR WRITE.

**Conditions**
Requires option 6-DBDDR3.
Requires a 6 Series MSO oscilloscope.
Set the DDR write reference level mode to Manual before using this command.
See Related Commands.

**Group**
Search and Mark

**Syntax**
```
```

**Related Commands**
```
SEARCH:SEARCH<x>:TRIGGER:A:DDRWRITE:REFLEVEL:MODE
```

**Arguments**

NR3 sets the DDR read search strobe mid reference value.

**Examples**

```
```
sets the DDR write search strobe mid reference value.

```
50E-3, indicating that the DDR write search strobe mid reference value is set
to 50E-3.
```

**SEARCH:SEARCH<x>:TRIGGER:A:DDRWRITE:REFLEVEL:MODE**

This command sets or queries the DDR write reference level mode to auto or
manual, when the search type is DDR WRITE.
Commands listed in alphabetical order

**Conditions**
Requires option 6-DB DDR3.
Requires a 6 Series MSO oscilloscope.

**Group**
Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode {AUTO|MANUAL}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode?
```

**Related Commands**
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REF Level:DATA:M ID
```

**Arguments**
- AUTO sets the DDR write reference level mode to auto.
- MANUAL sets the DDR write reference level mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.

**Examples**
```
```

**SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard**

This command sets or queries the DDR write search standard as DDR3 or LPDDR3.

**Conditions**
Requires option 6-DB DDR3
Requires 6 Series MSO oscilloscope

**Group**
Search and Mark

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard {DDR3|LPDDR3}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard?
```
Arguments

- DDR3 sets the DDR write search standard as DDR3.
- LPDDR3 sets the DDR write search standard as LPDDR3.

Examples


**SEARCH:SEARCH<x>:TRIGGER:A:DDRWRITE:STROBESOURCE**

This command sets or queries the DDR write strobe source when the search type is DDR Write.

Conditions

- Requires option 6-DBDDR3
- Requires 6 Series MSO oscilloscope
- Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGGER:A:DDRWRITE:STROBESOURCE
{CH<x>|CH<x>_D<x>|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGGER:A:DDRWRITE:STROBESOURCE?
```

Arguments

- CH<x> specifies channel <x> as the DDR write strobe source for the specified search <x>.
- CH<x>_D<x> specifies digital waveform <x> of channel <x> as the DDR write strobe source for the specified search <x>.
- Math<x> specifies math waveform <x> as the DDR write strobe source for the specified search <x>.
- REF<x> specifies reference waveform <x> as the DDR write strobe source for the specified search <x>.
- REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR write strobe source for the specified search <x>.

Examples

Commands listed in alphabetical order


**SEARCH:SEARCH<x>:TRIGGER:A:EDGE:SLOpe**

This command sets or queries the slope for an edge trigger search to determine where to place a mark. The search number is specified by <x>.

**Group**
Search and Mark

**Syntax**
SEARCH:SEARCH<x>:TRIGGER:A:EDGE:SLOpe \{RISe|FALL|EITher\}
SEARCH:SEARCH<x>:TRIGGER:A:EDGE:SLOpe?

**Arguments**
RISe specifies a rising edge.
FALL specifies a falling edge.
EITHER specifies either rising or falling edge.

**Examples**
SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE RISE sets the slope for search 1 to rise.

**SEARCH:SEARCH<x>:TRIGGER:A:EDGE:SOURce**

This command sets or queries the source waveform for an edge trigger search to determine where to place a mark. The search number is specified by <x>.

**Group**
Search and Mark

**Syntax**
SEARCH:SEARCH<x>:TRIGGER:A:EDGE:SOURce \{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>\}
SEARCH:SEARCH<x>:TRIGGER:A:EDGE:SOURce?

**Arguments**
CH<x> specifies one input channel as the edge source, where the channel number is specified by <x>. 

2-930 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
CH<x>_D<x> specifies a digital reference waveform as the source waveform for the specified search.

MATH<x> specifies the math waveform as the search source, where the math number is specified by <x>.

REF<x> specifies the reference waveform as the search source, where the reference number is specified by <x>.

REF<x>_D<x> specifies a digital reference waveform as the source waveform for the specified search.

Examples


SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold

This command sets or queries the source threshold level for an edge trigger search to determine where to place a mark. The search number is specified by <x>.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold?

Arguments
<NR3> is the source threshold level for an edge trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:EDGE:THReshold 50.0e-3 sets the threshold to 50 mV.

SEARCH:SEARCH1:TRIGGER:A:EDGE:THRESHOLD 0.0E+0 indicating the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:CLOCk:THReshold

This command sets or queries the logic clock threshold for a logic trigger search to determine where to place a mark. The search number is specified by <x>.
Commands listed in alphabetical order

**Group**  
Search and Mark

**Syntax**  
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:CLOCK:THReshold <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:CLOCK:THReshold?

**Arguments**  
<NR3> is the logic clock threshold.

**Examples**  
SEARCH:SEARCH1:TRIGger:A:LOGIC:CLOCK:THReshold 50.0e-3 sets the threshold to 50 mV.  
SEARCH:SEARCH1:TRIGger:A:LOGIC:CLOCK:THReshold? might return SEARCH:SEARCH1:TRIGger:A:LOGIC:CLOCK:THRESHOLD 0.0E+0 indicating the threshold is set to 0.0 V.

**SEARCH:SEARCH<x>:TRIGger:A:LOGIC:DELTatime**  
This command specifies the Logic search delta time value. The time value is used as part of the Logic search condition to determine if the duration of a logic pattern meets the specified time constraints. The search number is specified by <x>.

**Group**  
Search and Mark

**Syntax**  
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:DELTatime <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:DELTatime?

**Arguments**  
<NR3> is delta time value.

**Examples**  
SEARCH:SEARCH1:TRIGger:A:LOGIC:DELTatime 1.5e-9 sets the delta time to 1.5 ns.  

**SEARCH:SEARCH<x>:TRIGger:A:LOGIC:FUNCtion**  
This command sets or queries the logic operator for a pattern or state trigger search to determine where to place a mark. The search number is specified by <x>.

**Group**  
Search and Mark
Commands listed in alphabetical order

**Syntax**

```plaintext
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCtion {AND|NANd|NOR|OR}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCtion?
```

**Arguments**

- **AND** places a mark if all conditions are true.
- **NANd** places a mark if any of the conditions are false.
- **NOR** places a mark if all conditions are false.
- **OR** places a mark if any of the conditions are true.

**Examples**

```
SEARCH:SEARCH1:TRIGGER:A:LOGIC:FUNCTION AND
sets the trigger a logic function for search 1 to AND.

SEARCH:SEARCH1:TRIGGER:A:LOGIC:FUNCTION?
might return
SEARCH:SEARCH1:TRIGGER:A:LOGIC:FUNCTION NOR,
indicating that the logic function for search 1 is set to NOR.
```

**SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPUT:CLOCK:SOUrce**

This command specifies or queries the channel to use as the clock source for logic trigger. The search number is specified by `<x>`.

**Group**

Search and Mark

**Syntax**

```plaintext
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:INPUT:CLOCK:SOUrce {CH<x>|Ch<x>_D<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:INPUT:CLOCK:SOUrce?
```

**Arguments**

- Arguments are the possible input channels.

**Examples**

```
sets the clock source to channel 2.

might return
indicating the clock source is not defined.
```

**SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:CH<x>**

This command sets or queries the voltage level to use for logic trigger search. The search number is specified by `<x>`.
Commands listed in alphabetical order

Group Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:LOGIC:LEVEL:CH<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:LEVEL:CH<x>?

Arguments

<NR3> is the voltage level to use for logic trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:LOGIC:LEVEL:CH2 50.0e-3 sets the level to 50 mV.
SEARCH:SEARCH1:TRIGGER:A:LOGIC:LEVEL:CH2 0.0E+0 indicating the level is set to 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:LOGIC:LEVEL:MATH<x>

This command sets the voltage level to use for logic trigger search. The search number is specified by <x>.

Group Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:LOGIC:LEVEL:MATH<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:LEVEL:MATH<x>?

Arguments

<NR3> is the voltage level to use for logic trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:LOGIC:LEVEL:MATH1 50.0e-3 sets the level to 50.0 mV.
SEARCH:SEARCH1:TRIGger:A:LOGIC:LEVEL:MATH1? might return
SEARCH:SEARCH1:TRIGGER:A:LOGIC:LEVEL:MATH1 0.0E+0 indicating the level is set to 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:LOGIC:LEVEL:REF<x>

This command sets the voltage level to use for logic trigger search. The search number is specified by <x>.

Group Search and Mark
Commands listed in alphabetical order

**SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:LEVEL:REF<x>**

Syntax

```
SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:LEVEL:REF<x> <NR3>
SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:LEVEL:REF<x>? 
```

Arguments

- `<NR3>` is the voltage level to use for logic trigger search.

Examples

```
SEARCH:SEARCH1:TRIGGER:A:LOGIC:LEVEL:REF1 50.0e-3 sets the level to 50.0 mV.
SEARCH:SEARCH1:TRIGGER:A:LOGIC:LEVEL:REF1 0.0E+0 indicating the level is set to 0.0 V.
```

**SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:LOGICPattern:CH<x>**

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by `<x>`.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:LOGICPattern:CH<x> {H|L|X}
SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:LOGICPattern:CH<x>? 
```

Arguments

- `H` specifies triggering when the pattern is high.
- `L` specifies triggering when the pattern is low.
- `X` specifies triggering when the pattern is high or low.

Examples

```
SEARCH:SEARCH1:TRIGGER:A:LOGIC:LOGICPattern:CH2 H sets the channel 2 pattern to a high.
SEARCH:SEARCH1:TRIGGER:A:LOGIC:LOGICPattern:CH2 X indicating channel 2 is a don't care.
```

**SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:LOGICPattern:CH<x>_D<x>**

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by `<x>`.
Commands listed in alphabetical order

**Group**  
Search and Mark

**Syntax**  
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:LOGICPattern:CH<x>_D<x>  
{H|L|X}  
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:LOGICPattern:CH<x>_D<x>?

**Arguments**  
H specifies triggering when the pattern is high.  
L specifies triggering when the pattern is low.  
X specifies triggering when the pattern is high or low.

**Examples**  
SEARCH:SEARCH1:TRIGger:A:LOGIC:LOGICPattern:CH1_D1 H sets the channel pattern to a high.  

**SEARCH:SEARCH<x>:TRIGger:A:LOGIC:LOGICPattern:MATH<x>**

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by <x>.

**Group**  
Search and Mark

**Syntax**  
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:LOGICPattern:MATH<x>  
{H|L|X}  
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:LOGICPattern:MATH<x>?

**Arguments**  
H specifies triggering when the pattern is high.  
L specifies triggering when the pattern is low.  
X specifies triggering when the pattern is high or low.

**Examples**  
SEARCH:SEARCH1:TRIGger:A:LOGIC:LOGICPattern:MATH1 H sets the pattern to a high.  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:REF<x>  

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by <x>.

Group  
Search and Mark

Syntax  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:REF<x> {H|L|X}  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:REF<x>?

Arguments  
H specifies triggering when the pattern is high.  
L specifies triggering when the pattern is low.  
X specifies triggering when the pattern is high or low.

Examples  
SEARCH:SEARCH1:TRIGger:A:LOGIc:LOGICPattern:REF1 H sets the pattern to a high.  
SEARCH:SEARCH1:TRIGGER:A:LOGIC:LOGICPATTERN:REF1 X indicating the pattern is a don’t care.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:POLarity  

This command sets or queries the polarity for the clock channel when Use Clock Edge is set to Yes for Logic search type. The search number is specified by <x>.

Group  
Search and Mark

Syntax  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:POLarity  
{POSitive|NEGative|EITher}  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:POLarity?

Arguments  
POSitive specifies using the positive clock edge.  
NEGative specifies using negative clock edge.  
EITher specifies using either the positive or negative clock edge.
Examples

SEARCH:SEARCH1:TRIGger:A:LOGIc:POLarity NEGATIVE sets the polarity to negative.

SEARCH:SEARCH1:TRIGger:A:LOGIc:POLarity? might return
SEARCH:SEARCH1:TRIGGER:A:LOGIC:POLARITY POSITIVE indicating the polarity is set to positive.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:USEClockedge

This command specifies whether or not Logic search uses a clock source. The search number is specified by <x>.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:LOGIc:USEClockedge {OFF|ON|0|1}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:USEClockedge?

Arguments OFF specifies not to use the clock source.

ON specifies to use the clock source.

0 specifies not to use the clock source.

1 specifies to use the clock source.

Examples

SEARCH:SEARCH1:TRIGger:A:LOGIc:USEClockedge ON specifies to use the clock source.

SEARCH:SEARCH1:TRIGGER:A:LOGIC:USECLOCKEDGE 0 indicating not to use the clock source.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:WHEn

This command sets or queries the condition for generating an A or B logic search with respect to the defined input pattern.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:WHEN {TRUE|FALSE|MOREThan|LESSThan|EQUAL|UNEQUAL}
SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:WHEN?
Arguments

TRUE searches on an input value that is true.
FALSE searches on an input value that is false.
MOREthan searches on an input value that is greater than a set value.
LESSthan searches on an input value that is less than a set value.
EQual searches on an input value that is equal to a set value.
UNEQual searches on an input value that is not equal to a set value.

Examples

SEARCH:SEARCH1:TRIGGER:A:LOGIC:WHEN FALSE specifies a search on an input value that is false.
SEARCH:SEARCH1:TRIGGER:A:LOGIC:WHEN TRUE indicating a search on an input value that is true.

SEARCH:SEARCH<x>:TRIGGER:A:PULSEWidth:HIGHLimit

This command specifies the upper limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values. The search number is specified by <x>.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGGER:A:PULSEWidth:HIGHLimit <NR3>
SEARCH:SEARCH<x>:TRIGGER:A:PULSEWidth:HIGHLimit?

Arguments

<NR3> is the upper limit to use, in seconds, when searching for a pulse.

Examples

SEARCH:SEARCH1:TRIGGER:A:PULSEWidth:HIGHLimit 2.5e-9 sets the high limit to 2.5 ns.
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:HIGHLIMIT 2.0000E-9 indicating the pulse width high limit is 2.0 ns.

SEARCH:SEARCH<x>:TRIGGER:A:PULSEWidth:LOGICQUALification

This command specifies whether or not to use logic qualification for a pulse width search. The search number is specified by <x>.
Commands listed in alphabetical order

Group          Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOGICQUALification
{ON|OFF}
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOGICQUALification?

Arguments
ON specifies to use logic qualification.
OFF specifies not to use logic qualification.

Examples
SEARCH:SEARCH1:TRIGger:A:PULSEWidth:LOGICQUALification ON
turns on logic qualification.
SEARCH:SEARCH1:TRIGger:A:PULSEWidth:LOGICQUALification? might
return SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:LOGICQUALIFICATION
OFF indicating logic qualification is off.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOWLimit

This command specifies the lower limit to use, in seconds, when searching for
a pulse whose duration is inside or outside a range of two values. The search
number is specified by <x>.

Group          Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:LOWLimit <NR3>
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:LOWLimit?

Arguments  <NR3> is the lower limit to use, in seconds, when searching for a pulse.

Examples  SEARCH:SEARCH1:TRIGger:A:PULSEwidth:LOWLimit 0.5e-9 sets the
low limit to 0.5 ns.
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:LOWLIMIT 1.0000E-9
indicating the low limit is 2.0 ns.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity

This command specifies the polarity for a pulse width search. The search number
is specified by <x>.
Commands listed in alphabetical order

**Group**

Search and Mark

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:POLarity
{POSitive|NEGative}
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:POLarity?
```

**Arguments**

POSitive specifies positive polarity for a pulse width search.
NEGative specifies negative polarity for a pulse width search.

**Examples**

```
SEARCH:SEARCH1:TRIGger:A:PULSEwidth:POLarity NEGATIVE sets the polarity to negative.
SEARCH:SEARCH1:TRIGger:A:PULSEwidth:POLarity? might return
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:POLARITY POSITIVE indicating the polarity is positive.
```

**SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:SOUrce**

This command sets and queries the source for the pulse width search input. The search number is specified by `<x>`.

**Group**

Search and Mark

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:SOUrce
{CH<x>|CH<x>_D<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:SOUrce?
```

**Arguments**

Arguments are possible sources.

**Examples**

```
SEARCH:SEARCH1:TRIGger:A:PULSEwidth:SOUrce CH1 sets the source to channel 1.
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:SOURCENEGATIVE might return
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:SOURCENEGATIVE CH2 indicating channel 2 is the source.
```

**SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:THReshold**

Sets or queries the source threshold level for a pulse width trigger search to determine where to place a mark. The search number is specified by `<x>`.
Commands listed in alphabetical order

**Group**  
Search and Mark

**Syntax**  
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:THReshold?

**Arguments**  
<NR3> is the source threshold level for a pulse width trigger search.

**Examples**  
SEARCH:SEARCH1:TRIGger:A:PULSEwidth:THReshold 1.0e-9 sets to 1.0 ns.
SEARCH:SEARCH1:TRIGger:A:PULSEwidth:THReshold? might return SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:THRESHOLD 0.0E+0 indicating the threshold is 0.0 s.

**SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WHEn**

This command specifies to search for a pulse with a width (duration) that is less than, greater than, equal to, or unequal to a specified value (set using SEARCH:A:PULSEwidth:WIDTH), OR whose SEARCH:A:PULSEwidth:LOWLimit and SEARCH:A:PULSEwidth:HIGHLimit). The search number is specified by <x>.

**Group**  
Search and Mark

**Syntax**  
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WHEn {LESSthan|MOREthan| EQual|UNEQual|WIThin|OUTside}
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WHEn?

**Arguments**  
LESSthan causes a search when a pulse is detected with a width less than the time set by the SEARCH:A:PULSEwidth:WIDTH command.
MOREthan causes a search when a pulse is detected with a width greater than the time set by the SEARCH:A:PULSEwidth:WIDTH command.
EQual causes a search when a pulse is detected with a width equal to the time period specified in SEARCH:A:PULSEwidth:WIDTH within a ±5% tolerance.
UNEQual causes a search when a pulse is detected with a width greater than or less than (but not equal) the time period specified in SEARCH:A:PULSEwidth:WIDTH within a ±5% tolerance.
WIThin causes a search when a pulse is detected that is within a range set by two values.
OUTside causes a search when a pulse is detected that is outside of a range set by two values.

Examples

SEARCH:SEARCH1:TRIGger:A:PULSEwidth:WHen Outside causes a search when a pulse is detected that is outside the set range.

SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:WHEN WITHIN indicating that a search will occur when a pulse is detected that is within a set range.

**SEARCH:SEARCH<x>:TRIGger:A:RUNT:LOGICQUALification**

This command specifies whether or not to use logic qualification for a runt search. The search number is specified by <x>.

Group Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:RUNT:LOGICQUALification {ON|OFF}
SEARCH:SEARCH<x>:TRIGger:A:RUNT:LOGICQUALification?

Arguments

ON specifies to use logic qualification for a runt search.

OFF specifies not to use logic qualification for a runt search.

Examples

SEARCH:SEARCH1:TRIGger:A:RUNT:LOGICQUALification ON turns on logic qualification.

SEARCH:SEARCH1:TRIGGER:A:RUNT:LOGICQUALIFICATION OFF indicating logic qualification is off.

**SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity**

This command specifies the polarity for the runt search. The search number is specified by <x>.

Group Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity {POSitive|NEGative|EITher}
SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity?

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer 2-943
Arguments
POSitive specifies using positive polarity for the runt search.
NEGative specifies using negative polarity for the runt search.
EITHER specifies using either positive or negative polarity for the runt search.

Examples
SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY POSITIVE indicating the polarity is set to positive.

SEARCH:SEARCH<x>:TRIGGER:A:RUNT:SOURce

This command sets and queries the source for the Runt search input. The search number is specified by <x>.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGGER:A:RUNT:SOURce {CH<x>|REF<x>}

Arguments
Arguments are the available sources.

Examples
SEARCH:SEARCH1:TRIGGER:A:RUNT:SOURce CH1 sets the source to channel 1.
SEARCH:SEARCH1:TRIGGER:A:RUNT:SOURce CH2 indicating the source is set to channel 2.

SEARCH:SEARCH<x>:TRIGGER:A:RUNT:THReshold:HIGH

This command sets or queries the source threshold HIGH level for a runt trigger search to determine where to place a mark.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGGER:A:RUNT:THReshold:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGGER:A:RUNT:THReshold:HIGH?
Arguments <NR3> is the source threshold HIGH level for a runt trigger search.

Examples SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIGH 50.0E-3 sets the high threshold to 50 mV.
SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIGH? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIGH 0.0E+0 indicating the high threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGGER:A:RUNT:THRESHOLD:LOW
Sets or queries the source threshold LOW level for a runt trigger search to determine where to place a mark. The search number is specified by <x>.

Group Search and Mark


Arguments <NR3> is the source threshold LOW level for a runt trigger search.

Examples SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW 50.0e-3 sets the threshold to 50 mV.
SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW 0.0E+0 indicating the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGGER:A:RUNT:WHEN
This command sets or queries the condition setting for a runt trigger search to determine where to place a mark. The search number is specified by <x>.

Group Search and Mark

Arguments

LESS than argument sets the oscilloscope to search if the a runt pulse is detected with width less than the time set by the SEARCH:SEARCH<x>:TRIGGER:A:RUNT:WIDTH command.

MORE than argument sets the oscilloscope to search if the a runt pulse is detected with width more than the time set by the SEARCH:SEARCH<x>:TRIGGER:A:RUNT:WIDTH command.

EQUAL argument sets the oscilloscope to search when the pattern is true for a time period equal to the time period specified in SEARCH:SEARCH<x>:TRIGGER:A:RUNT:WIDTH within a ±5% tolerance.

NOTEQUAL argument sets the oscilloscope to search when the pattern is true for a time period greater than or less than (but not equal) the time period specified in SEARCH:SEARCH<x>:TRIGGER:A:RUNT:WIDTH within a ±5% tolerance.

OCCURS argument specifies a search event if a runt of any detectable width occurs.

Examples

SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEN MORE THAN sets the oscilloscope to trigger when a runt pulse is detected with width wider than the time set by the SEARCH:SEARCH<x>:TRIGGER:A:RUNT:WIDTH command.


SEARCH:SEARCH<x>:TRIGGER:A:RUNT:WIDTH

This command sets or queries the width setting for a runt trigger search to determine where to place a mark. The search number is specified by <x>.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGGER:A:RUNT:WIDTH <NR3>
SEARCH:SEARCH<x>:TRIGGER:A:RUNT:WIDTH?

Arguments

<NR3> specifies the minimum width in seconds.

Examples

SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH 400E–12 sets the runt trigger minimum width for search 1 to 0.4 nanoseconds.

SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH 500.0000E–12, indicating that the runt trigger minimum width for search 1 is set to 0.5 nanoseconds.
**SEARCH:SEARCH<x>:TRIGGER:A:SETHold:CLOCK:EDGE**

This command sets or queries the clock slope setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by `<x>`.

**Group** Search and Mark

**Syntax**

```
SEARCH:SEARCH<x>:TRIGGER:A:SETHold:CLOCK:EDGE {FALL|RISe}
SEARCH:SEARCH<x>:TRIGGER:A:SETHold:CLOCK:EDGE?
```

**Arguments**

- **FALL** specifies the polarity as the clock falling edge.
- **RISe** specifies the polarity as the clock rising edge.

**Examples**


**SEARCH:SEARCH<x>:TRIGGER:A:SETHold:CLOCK:SOURce**

This command sets or queries the clock source setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by `<x>`.

**Group** Search and Mark

**Syntax**

```
SEARCH:SEARCH<x>:TRIGGER:A:SETHold:CLOCK:SOURce {CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGGER:A:SETHold:CLOCK:SOURce?
```

**Arguments**

- **CH<x>** specifies an input channel as the edge source, where `<x>` = 1, 2, 3, 4, 5, 6, 7, or 8, depending on the number of channels in your instrument.
- **CH<x>_D<x>** specifies a digital waveform as the setup and hold clock source waveform for the specified search.
- **MATH<x>** specifies the math waveform as the search source, where `<x> = ≥1`.
- **REF<x>** specifies the reference waveform as the search source, where `<x> = ≥1`.
REF<\text{x}>_D<\text{x}> specifies a digital reference waveform as the setup and hold clock source waveform for the specified search.

Examples

\begin{verbatim}
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:SOURCE MATH1 sets the setup/hold trigger clock source setting for search 1 to MATH1.

\end{verbatim}

\textbf{SEARCH:SEARCH<\text{x}>:TRIGger:A:SETHold:CLOCK:THReshold}

This command sets or queries the clock threshold setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by <\text{x}>.

\textbf{Group}

Search and Mark

\textbf{Syntax}

\begin{verbatim}
SEARCH:SEARCH<\text{x}>:TRIGGER:A:SETHold:CLOCK:THReshold <NR3>
SEARCH:SEARCH<\text{x}>:TRIGGER:A:SETHold:CLOCK:THReshold?
\end{verbatim}

\textbf{Arguments}

<NR3> the clock threshold setting for a setup/hold trigger search.

\textbf{Examples}

\begin{verbatim}
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD -1.3 sets the setup/hold trigger clock threshold setting for search 1 to -1.3 volts.
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD -1.3000, indicating that the setup/hold trigger clock threshold setting for search 1 is set to -1.3 volts.
** CANNOT USE LocalCmd HERE BECAUSE OF VARIABLES **
\end{verbatim}

\textbf{SEARCH:SEARCH<\text{x}>:TRIGger:A:SETHold:HOLDTime}

This command sets or queries the hold time setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by <\text{x}>.

\textbf{Group}

Search and Mark

\textbf{Syntax}

\begin{verbatim}
SEARCH:SEARCH<\text{x}>:TRIGGER:A:SETHold:HOLDTime <NR3>
SEARCH:SEARCH<\text{x}>:TRIGGER:A:SETHold:HOLDTime?
\end{verbatim}

\textbf{2-948 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer}
Commands listed in alphabetical order

**Arguments**

<NR3> specifies the hold time setting in seconds. Positive values for hold time occur after the clock edge. Negative values occur before the clock edge.

**Examples**

SEARCH:SEARCH1:TRIGGER:A:SETHOLD:HOLDTIME 400E-12 sets the setup/hold trigger hold time setting for search 1 to 400 ps.

SEARCH:SEARCH1:TRIGGER:A:SETHOLD:HOLDTIME? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:HOLDTIME 500.0000E–12, indicating that the setup/hold trigger hold time setting for search 1 is set to 0.5 ns.

**SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:CH<x>**

This command sets or queries the voltage level to use for setup & hold trigger search. The search number is specified by <x>.

**Group**

Search and Mark

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:CH<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:CH<x>? 

**Arguments**

<NR3> the voltage level to use for setup & hold trigger search

**Examples**

SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:CH1 50.0e-3 sets the level to 50.0 mV.

SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:CH1? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LEVEL:CH1 0.0E+0 indicating the level is 0.0 V.

**SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:MATH<x>**

This command sets or queries the voltage level to use for setup & hold trigger search. The search number is specified by <x>.

**Group**

Search and Mark

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:MATH<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:MATH<x>? 

**Arguments**

<NR3> the voltage level to use for setup & hold trigger search.
Examples

SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:MATH1 50.0e-3 sets the level to 50.0 mV.

SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:MATH1? might return
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LEVEL:MATH1 0.0E+0 indicating the level is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:REF<x>

This command sets or queries the voltage level to use for setup & hold trigger search. The search number is specified by <x>.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:REF<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:REF<x>?

Arguments
<NR3> is the voltage level to use for setup & hold trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:REF1 50.0e-3 sets the level to 50.0 mV.

SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:REF1? might return
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LEVEL:REF1 0.0E+0 indicating the level is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by <x>.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>
{INCLude|DONTInclude}
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>?

Arguments
INCLude specifies including the specified channel SETHOLD inputs in the specified search.
DON'TInclude specifies not including the specified channel SETHOLD inputs in the specified search.

Examples
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LOGICPattern:CH1 INCLUDE specifies including the specified channel SETHOLD inputs in the specified search.


SEARCH:SEARCH<x>:TRIGGER:A:SETHOLD:LOGICPattern:CH<x>_D<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by <x>.

Group Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGGER:A:SETHOLD:LOGICPattern:CH<x>_D<x> {INCLUDE|DON'TInclude} SEARCH:SEARCH<x>:TRIGGER:A:SETHOLD:LOGICPattern:CH<x>_D<x>?

Arguments
INCLUDE specifies including the specified digital channel SETHOLD inputs in the specified search.

DON'TInclude specifies not including the specified digital channel SETHOLD inputs in the specified search.

Examples
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LOGICPattern:CH1_D0 INCLUDE specifies including the specified digital channel SETHOLD inputs in the specified search.


SEARCH:SEARCH<x>:TRIGGER:A:SETHOLD:LOGICPattern:MATH<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the
selected pattern may be true and still generate the trigger. The search number is specified by <x>.

**Group** Search and Mark

**Syntax**

```
SEARCH:SEARCH<x>:TRIGGER:A:SETHold:LOGICPattern:MATH<x>
{INCLUDE|DONTInclude}
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:MATH<x>?
```

**Arguments**

- **INCLUDE** specifies including the specified math SETHOLD inputs in the specified search.
- **DONTInclude** specifies not including the specified math SETHOLD inputs in the specified search.

**Examples**

```
SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:MATH1 INCLUDE
```

specifies including the specified math SETHOLD inputs in the specified search.

```
SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:MATH1?
```

might return "```
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LOGICPATTERN:MATH1
```
DONTINCLUDE indicating the specified math SETHOLD inputs will not be included in the specified search.

**SEARCH:SEARCH<x>:TRIGGER:A:SETHold:LOGICPattern:REF<x>**

This command sets and returns the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by <x>.

**Group** Search and Mark

**Syntax**

```
SEARCH:SEARCH<x>:TRIGGER:A:SETHold:LOGICPattern:REF<x>
{INCLUDE|DONTInclude}
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:REF<x>?
```

**Arguments**

- **INCLUDE** specifies including the specified reference SETHOLD inputs in the specified search.
- **DONTInclude** specifies not including the specified reference SETHOLD inputs in the specified search.
**Examples**


**SEARCH:SEARCH<x>:TRIGGER:A:SETHOLD:SETTime**

This command sets or queries the setup time setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by <x>.

**Group** Search and Mark

**Syntax**

SEARCH:SEARCH<x>:TRIGGER:A:SETHOLD:SETTime <NR3>

SEARCH:SEARCH<x>:TRIGGER:A:SETHOLD:SETTime?

**Arguments**

<NR3> specifies the setup time for setup and hold violation triggering.

**Examples**

SEARCH:SEARCH1:TRIGGER:A:SETHOLD:SETTIME 2E–9 sets the setup/hold trigger setup time setting for search 1 to 2 ns.


**SEARCH:SEARCH<x>:TRIGGER:A:STATE**

This command sets or queries the enabled state of the search. The search number is specified by <x>.

**Group** Search and Mark

**Syntax**

SEARCH:SEARCH<x>:TRIGGER:A:STATE {<NR1>|OFF|ON}

**Arguments**

<NR1> = 1 enables the search. Any other character disables the search.

ON enables the search.

OFF disables the search.
Examples
SEARCH:SEARCH1:TRIGGER:A:STATE OFF disables the search.
SEARCH:SEARCH1:TRIGGER:A:STATE? might return
SEARCH:SEARCH1:TRIGGER:A:STATE 1 indicating search 1 is enabled.

SEARCH:SEARCH<x>:TRIGGER:A:STOPAcq

This command sets or queries whether acquisitions are stopped when a search hit is found. The search number is specified by <x>.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGGER:A:STOPAcq {<NR1>|OFF|ON}
SEARCH:SEARCH<x>:TRIGGER:A:STOPAcq?

Arguments
<x> is the number of the search on which to enable or disable the stop acquisition function.

<NR1> = 1 enables stopping when a search hit is found. Any other character disables the feature.

ON enables stopping when a search hit is found.

OFF disables stopping on a search hit.

Examples
SEARCH:SEARCH1:TRIGGER:A:STOPAcq ON enables stopping when a search hit is found.
:SEARCH:SEARCH3:TRIGGER:A:STOPACQ 1 indicating acquisitions are stopped when a search hit is found on search number 3.

SEARCH:SEARCH<x>:TRIGGER:A:TIMEOut:LOGICQUALification

This command specifies whether or not to use logic qualification for a timeout search. The search number is specified by <x>.

Group
Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGGER:A:TIMEOut:LOGICQUALification {ON|OFF}
SEARCH:SEARCH<x>:TRIGGER:A:TIMEOut:LOGICQUALification?
Commands listed in alphabetical order

Arguments
ON specifies to use logic qualification.
OFF specifies not to use logic qualification.

Examples
SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:LOGICQUALIFICATION ON specifies to use logic qualification.

SEARCH:SEARCH<x>:TRIGGER:A:TIMEOUT:POLARITY
The polarity to be used for a Timeout search. The search number is specified by <x>.

Group Search and Mark

Syntax
SEARCH:SEARCH<x>:TRIGGER:A:TIMEOUT:POLARITY
{STAYSHigh|STAYSLow|Either}
SEARCH:SEARCH<x>:TRIGGER:A:TIMEOUT:POLARITY?

Arguments
STAYSHigh specifies the polarity stays HIGH.
STAYSLow specifies the polarity stays LOW.
Either specifies the polarity stays HIGH or stays LOW.

Examples
SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY STAYSLow specifies the polarity stays LOW.

SEARCH:SEARCH<x>:TRIGGER:A:TIMEOUT:SOURce
This command sets and queries the source for timeout search input. The search number is specified by <x>.

Group Search and Mark
Commands listed in alphabetical order

**SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce**

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce
{CH<x> | CH<x>_D<x> | MATH<x>REF<x> | REF<x>_D<x>}
```

Arguments

Arguments are the available sources.

Examples

```
SEARCH:SEARCH1:TRIGger:A:TIMEOut:SOUrce
SEARCH:SEARCH1:TRIGger:A:TIMEOut:SOUrce?
```

might return

```
SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:SOURCE CH1_D0
```

indicating

CH1_D0 is the source.

**SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold**

Sets or queries the source threshold level for a timeout trigger search to determine where to place a mark. The search number is specified by <x>.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold?
```

Arguments

<NR3> is the source threshold level for a timeout trigger search.

Examples

```
SEARCH:SEARCH1:TRIGger:A:TIMEOut:THReshold 50.0e-3 sets the
threshold to 50.0 mV.
SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:THRESHOLD 0.0E+0 indicating
the threshold is 0.0 V.
```

**SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIMe**

This command sets or queries the time setting for a timeout trigger search to determine where to place a mark. The search number is specified by <x>.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIMe <NR3>
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIMe?
```
Arguments

<NR3> is the time in seconds.

Examples

SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME 400E-9 sets the timeout trigger time setting for search 1 to 400 ns.

SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME 500.0000E-12, indicating that the timeout trigger time setting for search 1 is set to 500 ns.

**SEARCH:SEARCH<x>:TRIGGER:A:TRANSITION:DELTATime**

This command sets or queries the transition time setting for a transition trigger search to determine where to place a mark. The search number is specified by <x>.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGGER:A:TRANSITION:DELTATime <NR3>
SEARCH:SEARCH<x>:TRIGGER:A:TRANSITION:DELTATime?

Arguments

<NR3> specifies the transition time in seconds.

Examples

SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME 400E-9 sets the transition trigger time setting for search 1 to 400 ns.

SEARCH:SEARCH1:TRIGGER:A:TRANSITION:DELTATIME 500.0000E-12, indicating that the transition trigger time setting for search 1 is set to 500 ps.

**SEARCH:SEARCH<x>:TRIGGER:A:TRANSITION:LOGICQUALification**

This command specifies whether or not to use logic qualification for a transition search. The search number is specified by <x>.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGGER:A:TRANSITION:LOGICQUALification {ON|OFF}
SEARCH:SEARCH<x>:TRIGGER:A:TRANSITION:LOGICQUALification?
Arguments

ON specifies to use logic qualification for a transition search.

OFF specifies not to use logic qualification for a transition search.

Examples


SEARCH:SEARCH<x>:TRIGGER:A:TRANSITION:POLARITY

This command specifies the polarity for the transition search. The search number is specified by <x>.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGGER:A:TRANSITION:POLARITY
{POSitive|NEGative|EITHER}
SEARCH:SEARCH<x>:TRIGGER:A:TRANSITION:POLARITY?

Arguments

EITHER places a mark on a transition of either polarity.

NEGATIVE places a mark on a transition of negative polarity.

POSITIVE places a mark on a transition of positive polarity.

Examples


SEARCH:SEARCH<x>:TRIGGER:A:TRANSITION:SOURCE

This command sets and queries the source for the transition search input. The search number is specified by <x>.

Group

Search and Mark
Commands listed in alphabetical order

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:TRANSition:SOUrce
{CH<x>|MATH<x>|REF<x>}
SEARCH:SEARCH<x>:TRIGger:A:TRANSition:SOUrce?
```

**Arguments**

Arguments are the available sources.

**Examples**

```
SEARCH:SEARCH1:TRIGger:A:TRANSition:SOUrce CH1 sets the source to channel 1.
SEARCH:SEARCH1:TRIGger:A:TRANSition:SOUrce? might return
SEARCH:SEARCH1:TRIGger:A:TRANSITION:SOUrce CH2 indicating channel 2 is the source.
```

---

**SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:HIGH**

Sets or queries the source threshold HIGH level for a transition trigger search to determine where to place a mark. The search number is specified by <x>.

**Group**

Search and Mark

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:HIGH?
```

**Arguments**

<NR3> the source threshold HIGH level for a transition trigger search.

**Examples**

```
SEARCH:SEARCH1:TRIGger:A:TRANSition:THReshold:HIGH 50.0e-3 sets the high threshold to 50.0 mV.
SEARCH:SEARCH1:TRIGger:A:TRANSITION:THRESHOLD:HIGH 0.0E+0 indicating the high threshold is 0.0 V.
```

---

**SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:LOW**

Sets or queries the source threshold LOW level for a transition trigger search to determine where to place a mark. The search number is specified by <x>.

**Group**

Search and Mark

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:LOW?
```
Arguments

<NR3> is the source threshold LOW level for a transition trigger search.

Examples

SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:LOW -50.0e-3 sets the LOW threshold to -50.0 mV.


SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:LOW 0.0E+0 indicating the LOW threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGGER:A:TRANSITION:WHEN

This command sets or queries the condition setting for a transition trigger search to determine where to place a mark. The search number is specified by <x>.

Group
Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGGER:A:TRANSITION:WHEN {FASTERthan|SLOWERthan|EQUAL|UNEQUAL}

SEARCH:SEARCH<x>:TRIGGER:A:TRANSITION:WHEN?

Arguments

FASTERthan sets the trigger to occur when the transitioning signal is faster than the set volts/second rate.

SLOWERthan sets the trigger to occur when the transitioning signal is slower than the set volts/second rate.

EQUAL sets the trigger to occur when the transitioning signal is equal to the set volts/second rate.

UNEQUAL sets the trigger to occur when the transitioning signal is not equal to the set volts/second rate.

Examples

SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEN SLOWERTHAN sets the transition trigger condition setting for search 1 to SLOWERTHAN.


SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEN FASTERTHAN, indicating that the transition trigger condition setting for search 1 is set to FASTERTHAN.

SEARCH:SEARCH<x>:TRIGGER:A:TYPE

This command sets or queries the trigger type setting for a search to determine where to place a mark. The search number is specified by <x>.
Group: Search and Mark

Syntax:

SEARCH:SEARCH<x>:TRIGger:A:TYPe
{EDGE|RUNT|TRANSition|PULSEwidth|TIMEOut|LOGIc|
SETHold|WINdow|Bus|DDRRead|DDRREADWrite|DDRWrite}
SEARCH:SEARCH<x>:TRIGger:A:TYPe?

Arguments:

EDGE triggers when the source input signal amplitude crosses the specified level in the direction given by the slope.

**NOTE. Some trigger types are optional.**

RUNT triggers when a pulse crosses the first preset voltage threshold but does not cross the second preset threshold before recrossing the first. The thresholds are set with the SEARCH:SEARCH<x>:TRIGger:A:RUNT:HIGh and SEARCH:SEARCH<x>:TRIGger:A:RUNT:LOW THRESHOLD commands.

TRANSition triggers when a pulse crosses both thresholds in the same direction as the specified polarity and the transition time between the two threshold crossings is greater or less than the specified time delta.

PULSEwidth triggers on input signal source pulses that are inside or outside of the given time range specified by
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATrern:WHEn:LESSLimit and
The polarity is selected using the SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLArity command.

TIMEOut triggers on an input signal source that stays above, stays below, or stays either above or below the trigger level for a given time.

LOGIc specifies that a search occurs when specified conditions are met, and is controlled by the SEARCH:A:LOGIc commands.

SETHold triggers on a functional pattern combination of one to three data sources at the time of the clock transition.

WINdow triggers on an input signal source that enters or exits the horizontal band defined by the two trigger levels.

Bus specifies that a search occurs when a communications signal is found.

DDRRead triggers on a DDR Read event.

DDRREADWrite triggers on a DDR Read or Write event.

DDRWrite triggers on a DDR Write event.
Examples

SEARCH:SEARCH1:TRIGGER:A:TYPE RUNT sets the trigger type setting for search 1 to RUNT.

SEARCH:SEARCH1:TRIGGER:A:TYPE? might return SEARCH:SEARCH1:TRIGGER:A:TYPE EDGE, indicating that the trigger type setting for search 1 is set to EDGE.

SEARCH:SEARCH<x>:TRIGGER:A:WINDOW:CROSSIng

This command sets or queries the window trigger threshold crossing of the selected trigger Source. The threshold crossing selection is only effective when :TRIGGER:A:WINDOW:WHEN is INSIDE Greater or OUTSIDE Greater. The search number is specified by <x>.

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGGER:A:WINDOW:CROSSIng

{UPPer|LOWer|EITHER|NONE}

Arguments

UPPer if :TRIGGER:A:WINDOW:WHEN is INSIDE Greater, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (:TRIGGER:A:WINDOW:WIDTH) and then exits through the upper threshold. If :TRIGGER:A:WINDOW:WHEN is OUTSIDE Greater, the instrument triggers when the signal remains above the upper threshold for longer than the time limit (:TRIGGER:A:WINDOW:WIDTH) and then crosses downward through the upper threshold.

LOWer if :TRIGGER:A:WINDOW:WHEN is INSIDE Greater, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (:TRIGGER:A:WINDOW:WIDTH) and then exits through the lower threshold. If :TRIGGER:A:WINDOW:WHEN is OUTSIDE Greater, the instrument triggers when the signal remains below the lower threshold for longer than the time limit (:TRIGGER:A:WINDOW:WIDTH) and then crosses upwards through the lower threshold.

EITHER if :TRIGGER:A:WINDOW:WHEN is INSIDE Greater, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (:TRIGGER:A:WINDOW:WIDTH) and then exits through either the upper or lower threshold. If :TRIGGER:A:WINDOW:WHEN is OUTSIDE Greater, the instrument triggers when the signal remains either above the upper threshold or below the lower threshold for longer than the time limit (:TRIGGER:A:WINDOW:WIDTH) and then crosses a threshold.

NONE if :TRIGGER:A:WINDOW:WHEN is INSIDE Greater, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (:TRIGGER:A:WINDOW:WIDTH) without crossing through either the
upper or lower threshold. If :TRIGGER:A:WINDOW:WHEN is OUTSIDEGreater, the instrument triggers when the signal remains outside the upper and lower thresholds for longer than the time limit (:TRIGGER:A:WINDOW:WIDTH) without crossing through either the upper or lower threshold.

**Examples**

SEARCH:SEARCH1:TRIGGER:A:WINDOW:CROSSING LOWER sets the CROSSING to LOWER.


**SEARCH:SEARCH<x>:TRIGGER:A:WINDOW:LOGICQUALIFICATION**

This command specifies or queries whether or not to use logic qualification for a window search. The search number is specified by <x>.

**Group** Search and Mark

**Syntax**

SEARCH:SEARCH<x>:TRIGGER:A:WINDOW:LOGICQUALIFICATION {ON|OFF}
SEARCH:SEARCH<x>:TRIGGER:A:WINDOW:LOGICQUALIFICATION?

**Arguments**

ON specifies to use logic qualification for a window search.

OFF specifies not to use logic qualification for a window search.

**Examples**

SEARCH:SEARCH1:TRIGGER:A:WINDOW:LOGICQUALIFICATION ON turns ON logic qualification for a window search.


**SEARCH:SEARCH<x>:TRIGGER:A:WINDOW:POLARITY**

This command sets or queries the window trigger threshold crossing of the selected trigger Source. The search number is specified by <x>.

**Group** Search and Mark
Commands listed in alphabetical order

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:WINdow:POLarity

{UPPer|LOWer|EITher|NONE}

SEARCH:SEARCH<x>:TRIGger:A:WINdow:POLarity?

**Arguments**

**UPPer** specifies that the instrument triggers when the signal remains above the upper threshold for longer than the time limit and then crosses downward through the upper threshold.

**LOWer** specifies that the instrument triggers when the signal remains below the lower threshold for longer than the time limit and then crosses upwards through the lower threshold.

**EITher** specifies that the instrument triggers when the signal remains either above the upper threshold or below the lower threshold for longer than the time limit and then crosses a threshold.

**NONE** specifies that the instrument triggers when the signal remains outside the upper and lower thresholds for longer than the time limit without crossing through either the upper or lower threshold.

**Examples**

SEARCH:SEARCH1:TRIGger:A:WINdow:POLarity LOWER sets the polarity to lower.

SEARCH:SEARCH1:TRIGger:A:WINdow:POLarity? might return

SEARCH:SEARCH1:TRIGger:A:WINdow:POLarity UPPER indicating the polarity is set to upper.

**SEARCH:SEARCH<x>:TRIGger:A:WINdow:SOUrce**

This command sets and queries the source for the window search input. The search number is specified by <x>.

**Group**

Search and Mark

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:WINdow:SOURce

{CH<x>|MATH<x>|REF<x>}

SEARCH:SEARCH<x>:TRIGger:A:WINdow:SOURce?

**Arguments**

Arguments are the available sources.

**Examples**

SEARCH:SEARCH1:TRIGger:A:WINdow:SOURce CH2 sets the source to channel 2.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:HIGH

This command sets or queries the source threshold HIGH level for a transition trigger search to determine where to place a mark. The search number is specified by <x>.

**Group**  
Search and Mark

**Syntax**  
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:HIGH <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:HIGH?

**Arguments**  
<NR3> is the source threshold HIGH level for a transition trigger search

**Examples**  
SEARCH:SEARCH1:TRIGger:A:WINdow:THReshold:HIGH 50.0e-3 sets the high threshold to 50.0 mV.  
SEARCH:SEARCH1:TRIGger:A:WINdow:THReshold:HIGH? might return SEARCH:SEARCH1:TRIGger:A:WINdow:THRESHOLD:HIGH 0.0E+0 indicating the high threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:LOW

This command sets or queries the source threshold LOW level for a transition trigger search to determine where to place a mark. The search number is specified by <x>.

**Group**  
Search and Mark

**Syntax**  
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:LOW <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:LOW?

**Arguments**  
<NR3> the source threshold LOW level for a transition trigger search

**Examples**  
SEARCH:SEARCH1:TRIGger:A:WINdow:THReshold:LOW -50.0e-3 sets the low threshold to -50.0 mV.
SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:LOW 0.0E+0 indicating
the low threshold is 0.0 V.

**SEARCH:SEARCH<x>:TRIGger:A:WINdow:WHEn**

This command sets or queries the window search event. The search number is
specified by <x>.

**Group** Search and Mark

**Syntax**
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WHEn
{ENTERSwindow|EXITSwindow| INSIDEGreater|OUTSIDEGreater}
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WHEn?

**Arguments**
ENTERSwindow specifies a window search when the signal enters the window.

EXITSwindow specifies a window search when the signal exits the window.

OUTSIDEGreater specifies a search when the signal leaves the window defined
by the threshold levels for the time specified by Width.

INSIDEGreater specifies a search when the signal enters the window defined by
the threshold levels for the time specified by Width.

**Examples**
SEARCH:SEARCH1:TRIGger:A:WINdow:WHEn EXITSWINDOW specifies a
window search when the signal exits the window.

SEARCH:SEARCH1:TRIGger:A:WINdow:WHEn? might return
SEARCH:SEARCH1:TRIGger:A:WINdow:WHEn ENTERSWINDOW indicating a
window search when the signal enters the window.

**SEARCH:SEARCH<x>:TRIGger:A:WINdow:WIDth**

This command sets or queries the width setting for a window trigger search to
determine where to place a mark. The search number is specified by <x>.

**Group** Search and Mark

**Syntax**
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WIDth <NR3>
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WIDth?
Arguments  

<NR3> specifies the minimum width in seconds.

ECL specifies a preset high level of -1.3 V and TTL specifies a preset high level of +1.4 V.

Examples  

SEARCH:SEARCH1:TRIGGER:A:WINDOW:WIDTH 400E–12 sets the window trigger width setting for search 1 to 0.4 ns.


SEARCH:SEARCH1:TRIGGER:A:WINDOW:WIDTH 500.0000E–12, indicating that the window trigger width setting for search 1 is set to 0.5 ns.

SEARCH:SELEceted

This command sets or queries the selected search, for example SEARCH1. The search number is specified by <x>.

Group  

Search and Mark

Syntax  

SEARCH:SELECTED SEARCH1

Arguments  

SEARCH1 is the specified search.

Examples  

SEARCH:SELECTED SEARCH1 specifies a search 1 search.

SEARCH:SELECTED? might return :SEARCH:SELECTED SEARCH1 indicating search 1 is selected.

SEARCHTABLe (No Query Form)

This command adds or deletes a new search event table in an Option 5-WIN (Microsoft Windows 10 OS) TekExpress compliance testing application.

Group  

Search and Mark

Syntax  

SEARCHTABLE {ADDNew|DELETE} <qstring>

Arguments  

ADDNew adds a new search events table in the display area.

DELETE removes a displayed search events table from the display area.

<qstring> contains the name of the search table.
Examples

SEARCHTABLE:ADDNew "SEARCHTABLE1" adds the SEARCHTABLE1 in the display area.

SEARCHTABLE:DELETE "SEARCHTABLE1" removes the SEARCHTABLE1 from the display area.

SET? (Query Only)

This query-only command returns the commands that list the instrument settings, except for configuration information for the calibration values. You can use these commands to return the instrument to the state it was in when you made the SET? query. The SET? query always returns command headers, regardless of the setting of the HEADer command. This is because the returned commands are intended to be sent back to the instrument as a command string. The VERBose command can still be used to specify whether the returned headers should be abbreviated or full-length.

This command is identical to the *LRN? command.

Group

Miscellaneous

Syntax

SET?

Related Commands

HEADer

*LRN?

VERBose

Examples

SET? might return the following response:

:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;REPET 1;FASTACQ:STATE 0;APPLICATION:GPKNOB1:ACTIVE 0;APPLICATION:GPKNOB2:ACTIVE 0;APPLICATION:WINDOW:HEIGHT 236;WIDTH 640;APPLICATION:SCOPEAPP:STATE RUNNING;WINDOW FULLSCREEN;APPLICATION:EXTAPP:STATE NOTRUNNING;AUXOUT:SOURCE ATRIGGER;EDGE FALLING;CMDBATCH 1;HEADER 1;LOCK NONE;ROSC:SOURCE INTERNAL;VERBOSE 1;ALIAS:STATE 0;DISPLAY:CLOCK 1;COLOR:PALETTE NORMAL;MATHCOLOR DEFAULT;REFCOLOR DEFAULT;DISPLAY:FILTER SINX;FORMAT YT;GRATICULE FULL;INTENSITY:WAVEFORM 75.0000;AUTOBRIGHT 1;DISPLAY:PERSISTENCE OFF;STYLE VECTORS;TRIGBAR SHORT;TRIGT 1;VARPERSIST 500.0000E-3;PORT FILE;DIAG:LEVEL SUBSYS;TRIGGER:A:MODE AUTO;TYPE EDGE;LEVEL 0.0000;HOLDOFF:BY DEFAULT;TIME 1.5000E-6;TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;
SOCKETServer:ENAble

This command enables or disables the socket server which supports a telnet or other TCPIP socket connection to send commands and queries to the instrument. The default state is enabled.

Group  Miscellaneous
Syntax  SOCKETServer:ENAble {0|1|OFF|ON}
SOCKETServer:ENAble?

Arguments  1 enables the socket server. If the state is 0 (disabled) and this command is sent to enable the socket server when the port is in use by another service, then the error event code 221 (Settings conflict) is posted to the event queue and the socket server remains disabled. In this case, select a different port number and attempt to enable the socket server again.

0 disables the socket server.

ON enables the socket server.

OFF disables the socket server.

Examples  SOCKETSERVER:ENABLE On enables the socket server.
SOCKETSERVER:ENABLE? might return :SOCKETSERVER:ENABLE 1 indicating the socket server is enabled.

SOCKETServer:PORT

This command sets the TCPIP port for the socket server connection.

Group  Miscellaneous
Syntax  SOCKETServer:PORT <NR1>
SOCKETServer:PORT?

Arguments  <NR1> is the TCPIP port for the socket server connection.
If the socket server is enabled and the port specified is in use by another service, then the error event code 221 (Settings conflict) is posted to the event queue and the socket server remains in its current state (i.e. enabled/disabled and port address remain unchanged).
Similarly, if any sessions are active when the port is changed, this same error event may be posted to the event queue and the port will remain unchanged. In this case, exit all current sessions and send the :SOCKETServer:PORT command again.

**Examples**

SOCKETSERVER:PORT 4000 sets the socket server port number to 4000.

SOCKETSERVER:PORT? might return :SOCKETSERVER:PORT 4000 indicating that the port number is 4000.

**SOCKETServer:PROTOCOL**

This command sets or queries the protocol for the socket server.

**Group**

Miscellaneous

**Syntax**

SOCKETServer:PROTOCOL {Terminal|NONE}

SOCKETServer:PROTOCOL?

**Arguments**

Terminal specifies terminal protocol for the socket server. When set to Terminal, a session startup message is sent to the socket and a command prompt is provided.

NONE disables the terminal features, allowing the server to be used for raw socket transactions, such as with a VISA socket server. The default setting is NONE.

**Examples**

SOCKETSERVER:PROTOCOL NONE sets the protocol to none.

SOCKETSERVER:PROTOCOL? might return :SOCKETSERVER:PROTOCOL TERMINAL indicating the protocol is set to terminal.

**SRE**

The *SRE (Service Request Enable) command sets and queries the bits in the Service Request Enable Register. For more information, refer to Registers.

**Group**

Status and Error

**Syntax**

*SRE <NR1>

*SRE?
**Arguments**

<NR1> is a value in the range from 0 through 255. The binary bits of the SRER are set according to this value. Using an out-of-range value causes an execution error. The power-on default for SRER is 0 if *PSC is 1. If *PSC is 0, the SRER maintains the previous power cycle value through the current power cycle.

**Examples**

*SRE 48 sets the bits in the SRER to binary 00110000.

*SRE? might return 32, showing that the bits in the SRER have the binary value of 00100000.

---

**Related Commands**

*CLS
DESE
*ESE
*ESR?
EVENT?
EVMsg?
FACTory
*STB?
**SRE**

Returns  <NR1>

Examples  *STB? might return 96, showing that the SBR contains the binary value 01100000.

TEKSecure (No Query Form)

This command initializes, for the current user, both waveform and setup memories, overwriting any previously stored data.

Equivalent to invoking Teksecure from the Utility menu. This is a time-consuming operation (3 to 5 minutes) and the instrument is inoperable until the TekSecure operation is complete.

Group  Miscellaneous

Syntax  TEKSecure

Examples  TEKSECURE initializes both waveform and setup memories.

TIMe? (Query Only)

This command queries the time that the instrument displays.

Group  Miscellaneous

Syntax  TIMe?

Related Commands  DATE?

Returns  <QString> is a time in the form “hh mm ss” where hh refers to a two-digit hour number, mm refers to a two-digit minute number from 01 to 60, and ss refers to a two-digit second number from 01 to 60.

Examples  TIMe?? might return :TIME "14:05:17", indicating the current time is set to 2:05 p.m. and 17 seconds.
**TIIMe:ZONe**

This command sets the time zone to the one specified.

**Group**  
Miscellaneous

**Syntax**  
TIMe:ZONE <QString>  
TIMe:ZONE?

**Arguments**  
<QString> is a quoted string representing the desired time zone.

**Examples**  
TIME:ZONE "America/Yellowknife" sets the time zone to UTC delta -7.  
TIME:ZONE? might return :TIME:ZONE "America/Los_Angeles".

**TIIMe:ZONe:UTCDELTa**

This command sets or queries the time zone using the difference between the desired time zone and UTC.

**Group**  
Miscellaneous

**Syntax**  
TIMe:ZONE:UTCDELTa <NR3>  
TIMe:ZONE:UTCDELTa?

**Arguments**  
<NR3> is the specified number of hours difference between the desired time zone and UTC which is equivalent to GMT. The deltas supported are: -12.00, -11.00, -10.00, -9.30, -9.00, -8.30, -8.00, -7.00, -6.00, -5.00, -4.00, -3.30, -3.00, -2.00, -1.00, 0.0, 1.00, 2.00, 3.00, 3.30, 4.00, 4.30, 5.00, 5.30, 6.00, 6.30, 7.00, 8.00, 9.00, 9.30, 10.00, 10.30, 11.00, 11.30, 12.00

**Examples**  
TIME:ZONE:UTCDELTa -7.0e0 sets the time zone to America/Yellowknife.  

**TOTaluptime? (Query Only)**

Total number of hours the oscilloscope has been turned on since the NV memory was last programmed, usually during the initial manufacturing process.
Commands listed in alphabetical order

**Group**  
**Miscellaneous**

**Syntax**  
TOTa1uptime?

**Returns**  
The total number of hours the instrument has been turned on since the NV memory was last programmed.

**Examples**  
TOTALUPTIME? might return :TOTALUPTIME 756 indicating the up time is 756 minutes.

**TOUC1HSCRReen:CALibrate (No Query Form)**

This command launches the touchscreen calibration procedure. This command is equivalent to tapping the Calibrate Touchscreen control in the Utility->Self Test menu.

**Group**  
**Self Test**

**Syntax**  
TOUCHSCRReen:CALibrate STARt

**Arguments**  
START launches the touchscreen calibration procedure.

**Examples**  
TOUCHSCREEN:CALIBRATE START launches the touchscreen calibration procedure.

**TOUC1HSCRReen:STATe**

This sets or queries the enabled state of the touch screen. This command is equivalent to pushing the Touch Off button on the front panel.

To completely disable front panel operation, use the following two commands: LOCK ALL; :TOUCHSCREEN:STATE OFF. To re-enable the front panel, send these two commands: LOCK NONE; :TOUCHSCREEN:STATE ON. The commands must be sent in that order.

**Group**  
**Miscellaneous**

**Syntax**  
TOUCHSCRReen:STATe {0|1|OFF|ON}  
TOUCHSCRReen:STATe?
Commands listed in alphabetical order

Related Commands
- LOCk

Arguments
- 0 disables the touch screen.
- ON enables the touch screen.
- OFF disables the touch screen.

Examples
- TOUCHSCREEN:STATE OFF disables the touch screen.
- TOUCHSCREEN:STATE? might return :TOUCHSCREEN:STATE 1 indicating the touch screen is enabled.

*TRG (No Query Form)

Performs a group execute trigger on commands defined by *DDT.

Group
- Miscellaneous

Syntax
- *TRG

Related Commands
- *DDT

Examples
- *TRG immediately executes all commands that have been defined by *DDT.

TRIGger

This command forces a trigger event to occur. The query returns the current trigger parameters for the instrument.

Group
- Trigger

Syntax
- TRIGger FORCE
- TRIGger?

Arguments
- FORCE creates a trigger event. If TRIGger:STATE is set to READy, the acquisition will complete. Otherwise, this command will be ignored. This is equivalent to pressing the Force button on the front panel.
Examples

TRIGGER FORCE forces a trigger event to occur.
TRIGGER? returns the current trigger parameters for the instrument.

**TRIGger:{A|B|B:RESET}**

This command sets the A, B, or B Reset trigger level automatically to 50% of the range of the minimum and maximum values of the trigger input signal. The query returns current trigger parameters. The trigger level is the voltage threshold through which the trigger source signal must pass to generate a trigger event. This command is equivalent to pushing the LEVEL knob on the front panel.

Group

Trigger

Syntax

TRIGger:{A|B|B:RESET} SETLevel
TRIGger:{A|B|B:RESET}?

Arguments

SETLevel sets the trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.

Examples

TRIGger:A SETLEVEL sets the trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.
TRIGger:A? returns current trigger parameters.

**TRIGger:{A|B}:BUS:B<x>:ARINC429A:CONDition**

This command specifies a field or condition for an ARINC429 bus to trigger on. The bus number is specified by x.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:ARINC429A:CONDition
{SOW|LABEL|DATA|LABELANDDATA |EOW|ERROR}
TRIGger:{A|B}:BUS:B<x>:ARINC429A:CONDition?
Arguments

SOW specifies triggering on the first bit of a word.

LABEL1 specifies triggering on a matching label.

DATA specifies triggering on matching packet data field(s).

LABELANDDATA specifies triggering on a matching label and matching packet data field(s).

EOW specifies triggering on the 32nd bit of a word.

ERROR specifies triggering on a specified error condition.

NOTE. The type of error triggered on is specified by TRIGGER:{A|B}:BUS:B<x>:ARINC429A:ERTYPE.

Examples

TRIGGER:A:BUS:B1:ARINC429A:CONDITION DATA specifies triggering on packets that contain matching data field(s).

TRIGGER:A:BUS:B1:ARINC429A:CONDITION? might return SOW, indicating that the bus is triggering on the first bit of each word in the packet.

TRIGGER:{A|B}:BUS:B<x>:ARINC429A:DATA:HIVALue

This command sets or queries the high value when trigger on an ARINC429 data field. The bus number is specified by x. The trigger condition must be set to DATA or LABELANDDATA, and the data qualifier must be INrange or OUTFrange.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

TRIGGER:{A|B}:BUS:B<x>:ARINC429A:DATA:HIVALue <QString>

TRIGGER:{A|B}:BUS:B<x>:ARINC429A:DATA:HIVALue?

Arguments

<QString> is the label value.

NOTE. The size of the QString is dependent on the data field format selected using BUS:Bx:ARINC429A:DATAFORMat. Also, the stored QString is reset to its default value whenever the data field format is changed.
Examples

"XXXXXXXXXXXXXXXXX1000" sets the value to XXXXXXXXXXXXXXXX1000.

might return
"XXXXXXXXXXXXXXXXX", indicating that the value is
XXXXXXXXXXXXXXXXX.

TRIGGER:{A|B}:BUS:B<x>:ARINC429A:DATA:QUALifier

This command sets or queries the qualifier to be used when triggering on data in the DATA field for an ARINC429 bus signal. The bus number is specified by x. The trigger condition must be set to DATa or LABELANDDATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

TRIGGER:{A|B}:BUS:B<x>:ARINC429A:DATA:QUALifier {EQual|UNEQual|LESSthan|MOREthan |LESSEqual|MORSEEQual|INrange|OUTrange}
TRIGGER:{A|B}:BUS:B<x>:ARINC429A:DATA:QUALifier?

Arguments

Arguments are the available data qualifiers.

NOTE. The trigger qualifier only applies to the bits defined as the data field via the bus data field format specifier (using BUS:B<x>:ARINC429A:DATAFORMAT).

Examples


TRIGGER:A:BUS:B1:ARINC429A:DATA:QUALIFIER? might return EQUAL, indicating that the data qualifier is set to equal.

TRIGGER:{A|B}:BUS:B<x>:ARINC429A:DATA:VALue

This command sets or queries the low value when triggering on an ARINC429 data field. The bus number is specified by x. The trigger condition must be set to DATa or LABELANDDATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.
Group  Trigger

Syntax

TRIGGER:{A|B}:BUS:B<x>:ARINC429A:DATA:VALUE <QString>
TRIGGER:{A|B}:BUS:B<x>:ARINC429A:DATA:VALUE?

Arguments <QString> is the label value.

NOTE. The size of the QString is dependent on the data field format selected using BUS:B<x>:ARINC429A:DATAFORMAT. Also, the stored QString is reset to its default value whenever the data field format is changed.

Examples


TRIGGER:A:BUS:B1:ARINC429A:DATA:VALUE? might return "XXXXXXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXXXXXXXXXX.

TRIGGER:{A|B}:BUS:B<x>:ARINC429A:ERRTYPE

This command sets or queries the error type when triggering on an ARINC429 bus signal. The bus number is specified by x. The trigger condition must be set to ERRor.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group  Trigger

Syntax

TRIGGER:{A|B}:BUS:B<x>:ARINC429A:ERRTYPE {ANY|PARity|WORD|GAP}
TRIGGER:{A|B}:BUS:B<x>:ARINC429A:ERRTYPE?

Arguments

ANY sets the error type to match any of the other available error types.

PARity sets the error type to match on parity errors (parity value results in even parity count for a word).

WORD sets the error type to match on word errors (any unframed or unknown decode data).

GAP sets the error type to match on gap violations (less than 4 bits idle time between two packets on the bus).
Commands listed in alphabetical order

Examples

TRIGGER:A:BUS:B1:ARINC429A:ERRTYPE PARITY sets the error type to match on parity errors.


TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:HIVALue

This command sets or queries the high value when triggering on an ARINC429 label field. The bus number is specified by x. The trigger condition must be set to LABel, and the label qualifier must be INrange or OUTrange.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:HIVALue?

Arguments
QString is the label value.

Examples

TRIGGER:A:BUS:B1:ARINC429A:LABEL:HIVALUE? might return "XXXXXXXX", indicating that the value is XXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:QUALifier

This command sets or queries the qualifier to be used when triggering on label data for an ARINC429 bus signal. The bus number is specified by x. The trigger condition must be set to LABel or LABELANDDATA.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Trigger
**TRIGger:**{A|B}:BUS:B<x>:ARINC429A:LABEL:QUALifier {EQUAL|UNEQual|LESSthan|MOREthan |LESSEQual|MOREEQual|INrange|OUTrange}

**TRIGger:**{A|B}:BUS:B<x>:ARINC429A:LABEL:QUALifier?

**Arguments**

Arguments are the available data qualifiers.

**NOTE.** If the trigger condition is set to LABELANDDATA, the label qualifier will be locked to Equal until the trigger condition is changed again.

**Examples**


TRIGGER:A:BUS:B1:ARINC429A:LABEL:QUALIFIER? might return EQUAL, indicating that the label qualifier is set to equal.

**TRIGger:**{A|B}:BUS:B<x>:ARINC429A:LABEL:VALue

This command sets or queries the low value when triggering on an ARINC429 label field. The bus number is specified by x. The trigger condition must be set to LABEL or LABELANDDATA.

**Conditions**

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**

Trigger

**Syntax**

TRIGGER:{A|B}:BUS:B<x>:ARINC429A:LABEL:VALUE <QString>

TRIGGER:{A|B}:BUS:B<x>:ARINC429A:LABEL:VALUE?

**Arguments**

<QString> is the label value.

**Examples**


TRIGGER:A:BUS:B1:ARINC429A:LABEL:VALUE? might return "XXXXXXXX", indicating that the value is XXXXXXXX.
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SDI:VALue

This command sets or queries the label when triggering on an ARINC429 SDI field. The bus number is specified by x. The trigger condition must be set to DATA or LABELANDDATA, and the data format must be set to DATA.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax TRIGger:{A|B}:BUS:B<x>:ARINC429A:SDI:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SDI:VALue?

Arguments <QString> is the label value.

Examples TRIGGER:A:BUS:B1:ARINC429A:SDI:VALUE "X0" sets the value to X0.
TRIGGER:A:BUS:B1:ARINC429A:SDI:VALUE? might return "XX", indicating that the value is XX.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:SSM:VALue

This command sets or queries the label value when triggering on an ARINC429 SSM field. The bus number is specified by x. The trigger condition must be set to DATA or LABELANDDATA, and the data format must be set to DATA or SDIDATA.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax TRIGger:{A|B}:BUS:B<x>:ARINC429A:SSM:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SSM:VALue?
Arguments: <QString> is the label value.

**NOTE.** The SSM field is only present when the selected data field format is DATA or SDIDATA (using BUS:B<x>:ARINC429A:DATAFORMAT). Also, the stored QStringList is reset to its default value whenever the data field format is changed.

Examples: TRIGGER:A:BUS:B1:ARINC429A:SSM:VALUE "X0" sets the value to X0.
TRIGGER:A:BUS:B1:ARINC429A:SSM:VALUE? might return "XX", indicating that the value is XX.

**TRIGger:{A|B}:BUS:B<x>:AUDio:CONDITION**

This command sets the condition (word select, start of frame, or matching data) to be used when triggering on an audio bus signal. The bus number is specified by <x>.

Conditions: Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group: Trigger

Syntax: TRIGger:{A|B}:BUS:B<x>:AUDio:CONDITION {SOF|DATa}
TRIGger:{A|B}:BUS:B<x>:AUDio:CONDITION?

Arguments: 
SOF enables triggering on a word select or start of frame (depending on Audio Type).
DATa enables triggering on matching data.

Examples: TRIGger:A:BUS:B1:AUDio:CONDITION SOF sets the condition to start of frame.

**TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HITDMVALue**

This command sets or queries the binary data string for the high data word to be used when triggering on an TDM audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDITION.
Commands listed in alphabetical order

Group       Trigger

Syntax TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HITDmValue <QString>
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HITDmValue?

Arguments <QString> is the binary data string for the high data word to be used when triggering on an TDM audio bus signal.

Examples TRIGger:A:BUS:B1:AUDio:DATa:HITDmValue "1100" sets the high value to 1100.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HIVALue

This command sets the upper word value to be used when triggering on an audio bus signal. The trigger condition must be set to DATA using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

The bus number is specified by <x>.

Conditions Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group       Trigger

Syntax TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HIVALue?

Arguments <QString> is the upper word value to be used when triggering on an audio bus signal.

**TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:OFFSet**

This command sets the data offset value to be used when triggering on an audio bus signal. The trigger condition must be set to DATa using `TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition`.

The bus number is specified by `<x>`.

**Conditions**
Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

**Group**
Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:OFFSet <NR1>
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:OFFSet?
```

**Arguments**

`<NR1>` is the data offset value.

**Examples**

```
  :TRIGGER:A:BUS:B1:AUDIO:DATA:OFFSET 1 indicating the data offset value
  is 1.
```

**TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:QUALifier**

This command sets the qualifier to be used when triggering on an audio bus signal. The trigger condition must be set to DATa using `TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition`.

The bus number is specified by `<x>`.

**Conditions**
Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

**Group**
Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:QUALifier {LESSthan|MOREthan|EQUAL|UNEQual|LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:QUALifier?
```
Arguments

LESS than sets the qualifier to less than.

MORE than sets the qualifier to greater than.

Equal sets the qualifier to equal.

UNE Qual sets the qualifier to not equal.

LES SEqual sets the qualifier to less than or equal.

MORE Equal sets the qualifier to greater than or equal.

IN range sets the qualifier to in range.

OUT range sets the qualifier to out of range.

Examples


:TRI G ger:A:BUS:B1:AUD io:DAT a:QUAL i fier EQUAL indicating the

qual i fes is set to equal.


This command sets or queries the binary data string for the single or low data word to be used when triggering on an TDM audio bus signal. The trigger condition must be set to DAT a using TRI G ger:A:B|B:BUS:B<x>:AUD io:COND i tion.

Group Trigger


TRI G ger:A:B|B:BUS:B<x>:AUD io:DAT a:TD M VAL u e?

Arguments <QString> is the binary data string for the single or low data word to be used when triggering on an TDM audio bus signal.


TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:VALue

This command sets the lower word value to be used when triggering on an audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

The bus number is specified by <x>.

Conditions
Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:VALue?

Arguments
<QString> specifies the trigger data lower word.

Examples
TRIGger:A:BUS:B1:AUDio:DATa:VALue "11001100101" sets the data value to XXXXXXXXXXXXXX11001100101.


TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:WORD

This command sets the alignment of the data (left, right or either) to be used to trigger on an audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

The bus number is specified by <x>.

Conditions
Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:WORD {EITHER|LEFT|RIGHT}
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:WORD?
Arguments

**EITHER** aligns the trigger data to either left or right.

**LEFT** aligns the trigger data to the left.

**RIGHT** aligns the trigger data to the right.

Examples

TRIGger:A:BUS:B1:AUDio:DATa:WORD LEFT sets the word alignment to the left.


:TRIGGER:A:BUS:B1:AUDIO:DATA:WORD EITHER indicating the trigger data is aligned to either left or right.

**TRIGger:{A|B}:BUS:B<x>:CAN:CONDition**

This command sets the condition (start of frame, frame type, identifier, matching data, EOF, missing ACK field, bit-stuffing error) to be used when triggering on a CAN bus signal. The bus number is specified by <x>.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:CAN:CONDition

{SOF|FRAMEtype|IDentifier|DATa|IDANDDATA|EOF|ERROR|FDBITS}

TRIGger:{A|B}:BUS:B<x>:CAN:CONDition?

Arguments

**SOF** enables triggering on the start of frame.

**FDBITS** enables triggering on the values of the BRS and ESI bits in an FD packet.

**FRAMEtype** enables triggering on the type of frame.

**IDENTifier** enables triggering on a matching identifier.

**DATa** enables triggering on matching data.

**IDANDDATA** enables triggering on a matching identifier and matching data.

**EOF** enables triggering on the end of frame.

**ERROR** enables triggering on a specified error condition.

Examples


TRIGger:{A|B}:BUS:B<x>:CAN:DATa:DIRection

This command sets the data direction (read, write or “nocare”) to be used to search on a CAN bus signal. The trigger condition must be set to IDenti fi er (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by <x>.

Arguments

READ sets the CAN data direction to READ.
WRITE sets the CAN data direction to WRITE.
NOCARE sets the CAN data direction to either.

Examples


TRIGger:{A|B}:BUS:B<x>:CAN:DATa:OFFSet

This command sets or queries the data offset value, in bytes, to use when triggering on the CAN data field. The bus number is specified by x. The trigger condition must be set to DATA or IDANDDATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger
Syntax

TRIGger:{A|B}:BUS:B<x>:CAN:DATA:OFFSET <NR1>
TRIGger:{A|B}:BUS:B<x>:CAN:DATA:OFFSET?

Related Commands

BUS:B<x>:CAN:STANDARD
TRIGger (A|B) BUS B<x>:CAN:DATA:SIZe
TRIGger (A|B) BUS B<x>:CAN:DATA:VALue

Arguments

<NR1> is an integer whose minimum and default values are -1 (don't care), and the maximum is up to 7 (for CAN 2.0) or up to 63 (for ISO CAN FD and Non-ISO CAN FD).

The maximum is dependent on the number of bytes being matched and the CAN standard selected. Its value is calculated as [Absolute Maximum] - [Data Match Size]. For CAN 2.0, the absolute maximum is 8 bytes. For ISO CAN FD and Non-ISO CAN FD, the absolute maximum is 64 bytes. The minimum data match size is 1 byte, which produces the ranges listed above. Increasing the data match size above 1 byte will adjust the range of valid data offset values accordingly.

Examples


If the CAN standard is set for CAN 2.0, and the trigger data size is set to 3, the maximum value for the data offset will be 5 (8 - 3 = 5).

If the CAN standard is set for ISO CAN FD or Non-ISO CAN FD, and the trigger data size is set to 8, the maximum value for the data offset will be 56 (64 - 8 = 56).

TRIGger:{A|B}:BUS:B<x>:CAN:DATA:QUALifier

This command sets the qualifier (<, >, =, ≠, ≤, ≥) to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATA (using TRIGger (A|B) BUS B<x>:CAN:CONDITION). The bus number is specified by <x>.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:CAN:DATA:QUALifier  
{LESSthan|MOREthan|EQUAL|UNEQual|LESSEQual|MORREQual}
Commands listed in alphabetical order

**TRIGger:{A|B}:BUS:B<x>:CAN:DATa:QUALifier?**

### Arguments
- **LESSthan** sets the oscilloscope to trigger when the data is less than the qualifier value.
- **MOREthan** sets the oscilloscope to trigger when the data is greater than the qualifier value.
- **EQUAL** sets the oscilloscope to trigger when the data is equal to the qualifier value.
- **UNEQUAL** sets the oscilloscope to trigger when the data is not equal to the qualifier value.
- **LESSEQUAL** sets the oscilloscope to trigger when the data is less than or equal to the qualifier value.
- **MOREEQUAL** sets the oscilloscope to trigger when the data is greater than or equal to the qualifier value.

### Examples
- **TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER LESSTHAN** sets the oscilloscope to trigger when the data is less than the qualifier value.
- **TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER?** might return **:TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER LESSTHAN**, indicating that the oscilloscope is set to trigger when the data is less than the qualifier value.

**TRIGger:{A|B}:BUS:B<x>:CAN:DATa:SIZe**

This command sets the length of the data string, in bytes, to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDDATA or DATA (using **TRIGger (A|B) BUS B<x>:CAN:CONDITION**). The bus number is specified by <x>.

### Conditions
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

### Group
Trigger

### Syntax
- **TRIGger:{A|B}:BUS:B<x>:CAN:DATa:SIZe <NR1>**
- **TRIGger:{A|B}:BUS:B<x>:CAN:DATa:SIZe?**

### Arguments
- **<NR1>** is the length of the data string in bytes.
Examples

:TRIGGER:A:BUS:B1:CAN:DATa:SIZE 1 indicating the data size is set to 1 byte.

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:VALue

This command sets the binary data value to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATa (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by <x>.

Conditions
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:VALue?

Arguments
QString is the data value in binary format. The only allowed characters in the QString are 0, 1, and X.

Examples
:TRIGGER:A:BUS:B1:CAN:DATA:VALUE "XXXXXXXX" indicating the data value is set to XXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:CAN:ERRType

This command sets or queries the type of error condition for a CAN bus to triggering on. The bus number is specified by x. The trigger condition must be set to ERROR.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Trigger
Commands listed in alphabetical order

**Syntax**

TRIGger:{A|B}:BUS:B<x>:CAN:ERRType
{ACKMISS|BITSTUFFing|FORMERRor|ANYERRor}

TRIGger:{A|B}:BUS:B<x>:CAN:ERRType?

**Arguments**

ACKMISS specifies triggering on a missing ACK field.

BITSTUFFing specifies triggering on a bit stuffing error.

FORMERRor specifies triggering on a CAN FD form error. To use this option, the CAN standard must be set to FDISO or FDNONISO.

ANYERRor specifies triggering on any error type.

**Examples**


TRIGGER:A:BUS:B1:CAN:ERRTYPE? might return ANYERROR, indicating that the bus is triggering on all error types.

**TRIGger:{A|B}:BUS:B<x>:CAN:FD:BRSBit**

This command sets or queries the value of the bit rate switch bit (BRS bit) for a CAN bus to triggering on. The bus number is specified by x. The trigger condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

**Conditions**

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**

Trigger

**Syntax**

TRIGger:{A|B}:BUS:B<x>:CAN:FD:BRSBit {ONE|ZERo|NOCARE}

TRIGger:{A|B}:BUS:B<x>:CAN:FD:BRSBit?

**Arguments**

ONE filters CAN FD packets to only match those where the BRS bit has a value of 1 (fast data enabled).

ZERo filters CAN FD packets to only match those where the BRS bit has a value of 0 (fast data disabled).

NOCARE disables filtering of CAN FD packets on the BRS bit.

**Examples**


TRIGGER:{A|B}:BUS:B<x>:CAN:FD:ESIBit

This command sets or queries the value of the error state indicator bit (ESI bit) for a CAN bus to triggering on. The bus number is specified by x. The trigger condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

**Conditions**
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**
Trigger

**Syntax**
TRIGGER:{A|B}:BUS:B<x>:CAN:FD:ESIBit {ONE|ZERO|NOCARE}
TRIGGER:{A|B}:BUS:B<x>:CAN:FD:ESIBit?

**Arguments**
- **ONE** filters CAN FD packets to only match those where the ESI bit has a value of 1 (recessive).
- **ZERO** filters CAN FD packets to only match those where the ESI bit has a value of 0 (dominant).
- **NOCARE** disables filtering of CAN FD packets on the ESI bit.

**Examples**

TRIGGER:{A|B}:BUS:B<x>:CAN:FRAMEtype

This command sets the frame type (data, remote, error or overload) to be used when triggering on a CAN bus signal. The trigger condition must be set to FRAMEtype (using TRIGGER:{A|B}:BUS:B<x>:CAN:CONDition). B<x> is the bus number. The bus number is specified by <x>.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.
Group | Trigger
---|---
Syntax | TRIGGER:{A|B}:BUS:B<x>:CAN:FRAMETYPE
{DATA|REMOTE|ERROR|OVERLOAD}
TRIGGER:{A|B}:BUS:B<x>:CAN:FRAMETYPE?
Arguments | DATA specifies a data frame type.
REMOTE specifies a remote frame type.
ERROR specifies an error frame type.
OVERLOAD specifies an overload frame type.
Examples | TRIGGER:A:BUS:B1:CAN:FRAMETYPE DATA sets the CAN trigger frame type to DATA.

**TRIGGER:{A|B}:BUS:B<x>:CAN:IDENTIFIER:MODE**

This command sets the addressing mode (standard or extended format) to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATA (using TRIGGER:{A|B}:BUS:B<x>:CAN:CONDITION). The bus number is specified by <x>.

Conditions | Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.
Group | Trigger
Syntax | TRIGGER:{A|B}:BUS:B<x>:CAN:IDENTIFIER:MODE
{STANDARD|EXTENDED}
TRIGGER:{A|B}:BUS:B<x>:CAN:IDENTIFIER:MODE?
Arguments | STANDARD specifies the standard addressing mode.
EXTENDED specifies the extended addressing mode.
Commands listed in alphabetical order

TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:VALUE

This command sets the binary address value to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATA (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by <x>.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:VALUE <QString>
TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:VALUE?

Arguments <QString> is up to 29 bits specifying the binary identifier value. The only allowed characters in the QString are 0, 1, and X.


TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

This command specifies a field or condition within an Ethernet frame to trigger on. The bus number is specified by <x>.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger
Syntax

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition
{SFD|MACADDRess|MACLENgth|IPHeader|TCPHeader|DATa|EOP|
IDLe|FCSError|QTAG}

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition?

Related Commands

Most of the other TRIGger:A:BUS:B<x>:ETHERnet commands are impacted by the setting of this command.

Arguments

SFD — Start of frame delimiter.
MACADDRess — MAC addresses field.
MACLENgth — MAC length/type field.
IPHeader — IP header field. This argument is only available when PROTOCOL is set to IPv4.
TCPHeader — TCP header field. This argument is only available when PROTOCOL is set to IPv4.
DATa — TCP/IPv4 or MAC protocol client data field. If the protocol is set to OTHER, then DATa refers to the MAC client data.
EOP — End of Packet field.
IDLe — Idle field.
FCSError — Frame Check Sequence Error (CRC) field.
QTAG — IEEE 802.1Q (VLAN) control information field. In order to use QTAG as a trigger condition, the frame type must be set to QTAG).

Examples


TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:HIVALue

When the Ethernet trigger condition is set to DATa, and the qualifier is set to either INrange or OUTrange, this command specifies the upper data value of the range. (Use the command TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:V ALue to specify the lower limit of the range.) The default is all X's (don't care). The bus number is specified by <x>.

Conditions

Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.
Commands listed in alphabetical order

Group: Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:HIVALue?

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe
TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

Arguments

<QString> is a quoted string whose length varies depending on the size setting, up to 32 bits. (Use the command TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe to specify the size.) The allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

"XXXXXXXXXXXXXXXXXXXXXXXX00001000" sets the upper limit of the range to XXXXXXXXXXXXXXXXXXXXXXXXXXX00001000 (when the trigger condition is set to DATa, and the qualifier is set to INrange or OUTrange).

TRIGger:A:BUS:B1:ETHERnet:DATa:HIVALue?
might return "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:OFFSet

When the Ethernet trigger condition is set to DATa, this command specifies where in the data field to look for the data trigger value. It specifies the offset into the data field, in bytes, where the value will be matched. The default is -1 (don’t care). The bus number is specified by <x>.

Conditions

Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group: Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:OFFSet <NR1>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:OFFSet?

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition
Commands listed in alphabetical order

**TRIGger\{A|B\}:BUS:B<x>:ETHERnet:DATa:VALue**

**Arguments**

\(<\text{NR1}\>\) is an integer whose minimum and default values are -1 (don't care) and maximum is 1,499.

**Examples**


TRIGger:A:BUS:B1:ETHERnet:DATa:OFFSet? might return -1, indicating that the data offset value is the default value, -1, meaning "don't care".

**TRIGger\{A|B\}:BUS:B<x>:ETHERnet:DATa:QUALifier**

This command sets or queries the qualifier to be used when triggering on an Ethernet bus signal. The trigger condition must be set to DATa. The bus number is specified by \(<x>\).

**Conditions**

Requires option 5-SREN or SUP5-SREN Triggering and Analysis application.

**Group**

Trigger

**Syntax**

TRIGger\{A|B\}:BUS:B<x>:ETHERnet:DATa:QUALifier \{EQUAL|UNEQUAL|LESSthan|MOREthan|LESSEQual|MORREEqual|INrange|OUTrange\}

TRIGger\{A|B\}:BUS:B<x>:ETHERnet:DATa:QUALifier?

**Arguments**

LESSthan sets the qualifier to less than.

MOREthan sets the qualifier to greater than.

EQUAL sets the qualifier to equal.

UNEQUAL sets the qualifier to not equal.

LESSEQual sets the qualifier to less than or equal.

MOREEQual sets the qualifier to greater than or equal.

INrange sets the qualifier to in range.

OUTrange sets the qualifier to out of range.

**Examples**

TRIGger:A:BUS:B1:ETHERnet:DATa:QUALifier LESSthan sets the qualifier to less than.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe

When the Ethernet trigger condition is set to DATA, this command specifies the number of contiguous TCP/IPv4/MAC client data bytes to trigger on. The bus number is specified by <x>.

Conditions Requires option 5-SREN or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZE <NR1>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZE?

Related Commands TRIGger:{A|B} BUS B<x>:ETHERnet:CONDition

Arguments The minimum and default values are 1 and maximum is 16, except when the qualifier is set to Inside Range or Outside Range. In these cases, the maximum size is 4.

TRIGger:A:BUS:B1:ETHERnet:DATa:SIZe? might return 6, indicating that the oscilloscope is set to trigger on 6 contiguous data bytes.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue

When the Ethernet trigger condition is set to DATA, and the qualifier is set to LESSthan, MOREthan, EQUAL, UNEQual, LESSEQual or MOREEQual, this command specifies the value to trigger on. When the Ethernet trigger condition is set to DATA, and the qualifier is set to INrange or OUtrange, this command specifies the lower limit of the range. (Use the command TRIGger:{A|B} BUS B<x>:ETHERnet:DATa:HIVALue to set the upper limit of the range.) The default is all X's (don’t care). The bus number is specified by <x>. 
Conditions
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue?

Related Commands
TRIGger:(A|B):BUS:B<x>:ETHERnet:CONDition
TRIGger:(A|B):BUS:B<x>:ETHERnet:DATa:OFFSet
TRIGger:(A|B):BUS:B<x>:ETHERnet:DATa:SIZe
TRIGger:(A|B):BUS:B<x>:ETHERnet:DATa:HIVALue

Arguments
QString is a quoted string where the allowable characters are 0, 1, and X. The allowable number of characters depends on the setting for size (using TRIGger:A:BUS:B<x>:ETHERnet:DATa:SIZe). The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples
TRIGger:A:BUS:B1:ETHERnet:DATa:VALue "00001000" sets the binary data to trigger on to 00001000, assuming the qualifier is set to LESSthan, MOREthan, Equal, UNEQual, LESSEqual or MOREEQual, and DATa:SIZe is set to 1 byte.

TRIGger:A:BUS:B1:ETHERnet:DATa:VALue "00001000" sets the lower limit of the range to 00001000, assuming the qualifier is set to INrange or OUTrange, and DATa:SIZe is set to 1 byte.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:VALue

When the Ethernet trigger condition is set to IPHeader, this command specifies the value of the 32-bit destination address that is to be used in the trigger (along with the source address and protocol value). The IP destination address is a standard IP address such as 192.168.0.1. The default is all X's (don't care). The bus number is specified by <x>.

Conditions
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group
Trigger
Syntax

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:VALue?

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDITION
TRIGger:{A|B}:BUS:B<x>:ETHERnet:SOURceaddr:VALue
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue

Arguments

<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:IPHeader:DESTinationaddr:VALue "00011001001000010110100000000001" sets the IP destination address to trigger on to 192.168.0.1.


TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue

When the Ethernet trigger condition is set to IPHeader, this command specifies the value of the 8-bit protocol field that is to be used in the trigger (along with the source and destination addresses). The default is all X’s (don’t care). The bus number is specified by <x>.

NOTE. Commonly used protocol values are 1 (ICMP), 2 (IGMP), 6 (TCP) and 17 (UDP).

Conditions

Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group

Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue?

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDITION
**Commands listed in alphabetical order**

**TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:SOURceaddr:VALUE**

**Arguments**  
<str> is a quoted string of up to 8 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

**Examples**  
TRIGger:A:BUS:B1:ETHERNET:IPHEADER:PROTOCOL:VALUE "01010010" would set the value to be used in the trigger to 01010010.


**TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:VALUE**

When the Ethernet trigger condition is set to IPHeader, this command specifies the value of the 32-bit source address that is to be used in the trigger (along with the destination address and protocol value). The IP source address is a standard IP address such as 192.168.0.1. The default is all X’s (don’t care). The bus number is specified by <x>.

**Conditions**  
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**  
Trigger

**Syntax**  
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:SOURceaddr:VALUE <str>

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:SOURceaddr:VALUE?

**Related Commands**  
TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDITION

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:VALUE

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOCOL:VALUE

**Arguments**  
<str> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.
Examples
TRIGger:A:BUS:B1:ETHERnet:IPHeader:SOURceaddr:VALue "000110010100001011010000000001" sets the IP source address to trigger on to 192.168.0.1.

TRIGger:A:BUS:B1:ETHERnet:IPHeader:SOURceaddr:VALue might return "XXXXXXXXXXXXXXXXXXXXXXXXXXXX01".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:DESTination:VALue

When the Ethernet trigger condition is set to MAC ADDRess, this command specifies the 48–bit MAC destination address that is to be used in the trigger (along with the source address value). The default is all X’s (don’t care). The bus number is specified by <x>.

**NOTE.** MAC Addresses are 48–bit values such as 08:00:11:1E:C9:AE hex.

Conditions
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:DESTination:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:DESTination:VALue?

Related Commands
TRIGger (A|B):BUS:B<x>:ETHERnet:CONDition

Arguments
<QString> is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDReSS:SOUrce:VALue

When the Ethernet trigger condition is set to MACADDress, this command specifies the 48–bit MAC source address value that is to be used in the trigger (along with the destination address value). The default is all X’s (don’t care). The bus number is specified by <x>.

**NOTE.** MAC Addresses are 48–bit values such as 08:00:11:1E:C9:AE hex.

**Conditions**
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**
Trigger

**Syntax**
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDReSS:SOUrce:VALue
<QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDReSS:SOUrce:VALue?

**Related Commands**
TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDReSS:DESTination:VALue

**Arguments**
<QString> is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

**Examples**
"XXXXXXXXXX010101111111000000011110101010101000" would set the MAC destination address to trigger on to XX:35:FC:07:AA:C8 hex.


TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:HIVALue

When the Ethernet trigger condition is set to MACLENgth, and the qualifier is set to INrange or OUtRange, this command specifies the upper data value of the range. (Use the command TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:V ALue to specify the lower limit of the range.) The default is all X’s (don’t care). The bus number is specified by <x>. 

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer 2-1005
<table>
<thead>
<tr>
<th>Conditions</th>
<th>Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Trigger</td>
</tr>
<tr>
<td>Syntax</td>
<td>TRIGger:{A</td>
</tr>
<tr>
<td></td>
<td>TRIGger:{A</td>
</tr>
<tr>
<td>Related Commands</td>
<td>TRIGger (A</td>
</tr>
<tr>
<td></td>
<td>TRIGger (A</td>
</tr>
<tr>
<td>Arguments</td>
<td>&lt;QString&gt; is a quoted string of up toe 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.</td>
</tr>
<tr>
<td></td>
<td>&quot;XXXXXXXX00001000&quot; sets the upper limit of the range to the hexadecimal value XX08 (when the trigger condition is set to MACLENgth, and the qualifier is set to INrange or OUTrange).</td>
</tr>
<tr>
<td></td>
<td>&quot;XXXXXXXX00001000&quot;.</td>
</tr>
</tbody>
</table>

**TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:VALue**

When the Ethernet trigger condition is set to MACLENgth, and the qualifier is set to LESSthan, MOREthan, EQUAL, UNEQUAL, LESSEEQual or MOREEQUAL, this command specifies the 16-bit value to trigger on. When the qualifier is set to INrange or OUTrange, this command specifies the lower limit of the range. (Use the command TRIGger (A|B):BUS B<x>:ETHERnet MAC:LENgth HIVALue to set the upper limit of the range.) The default is all X’s (don’t care). The bus number is specified by <x>.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Trigger</td>
</tr>
<tr>
<td>Syntax</td>
<td>TRIGger:{A</td>
</tr>
<tr>
<td></td>
<td>TRIGger:{A</td>
</tr>
</tbody>
</table>
Commands listed in alphabetical order

**Related Commands**

- `TRIGger:(A|B):BUS:B<x>:ETHERnet:CONDition`

**Arguments**

- `<QString>` is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

**Examples**

  "XXXXXXXX00001000"` sets the MAC length/type value to trigger on the hexadecimal value XX08, assuming the qualifier is set to LESSthan, MOREthan, EQUAL, UNEQual, LESSEQual or MOREEQual.
  "XXXXXXXX00001000"` sets the lower limit of the range to the hexadecimal value XX08, assuming the qualifier is set to INrange or OUTrange.

**TRIGger:{A|B}:BUS:B<x>:ETHERnet:QTAG:VALue**

When the Ethernet trigger condition is set to QTAG, this command specifies the 32-bit Q-Tag value to trigger on. The default is all X’s (don’t care). The bus number is specified by `<x>`.

**Conditions**

Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**

Trigger

**Syntax**

- `TRIGger:{A|B}:BUS:B<x>:ETHERnet:QTAG:VALue <QString>`
- `TRIGger:{A|B}:BUS:B<x>:ETHERnet:QTAG:VALue?`

**Related Commands**

- `TRIGger:(A|B):BUS:B<x>:ETHERnet:CONDition`

**Arguments**

- `<QString>` is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

**Examples**

  "XXXXXXXXXXXXXXXXXXXX010010001010"` would specify to trigger on the Q-Tag value of hexadecimal XXXXX48A.

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer 2-1007
Commands listed in alphabetical order


TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue

When the Ethernet trigger condition is set to TCPHeader, this command specifies the 32-bit acknowledgement number that is to be used in the trigger (along with the destination and source port addresses and the sequence number). The default is all X's (don't care). The bus number is specified by <x>.

Conditions
Requires option 5-SREN or SUP5-SREN Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue
<QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue?

Related Commands
TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDITION
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue

Arguments
<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples
"xxxxxxxxxxxxxxxxxxxxxxxx00001000" sets the acknowledgement number to be used in the trigger to hexadecimal xxxxx08.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue

When the Ethernet trigger condition is set TCPHeader, this command specifies the 16-bit destination port address value that is to be used in the trigger (along with the acknowledgement value, source port address and the sequence number). The default is all X’s (don’t care). The bus number is specified by <x>.

Conditions
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue?

Related Commands
TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue

Arguments
<QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples
TRIGger:A:BUS:B1:ETHERnet:TCPHeader:DESTinationport:VALue "XXXXXXXX00100010" would set the destination port address value that is to be used in the trigger to hexadecimal XX22.
might return "XXXXXXXXXXXXXXXXX".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue

When the Ethernet trigger condition is set to TCPHeader, this command specifies the 32-bit sequence number that is to be used in the trigger (along with the destination and source port addresses and the acknowledgement value). The default is all X’s (don’t care). The bus number is specified by <x>.
Commands listed in alphabetical order

**Conditions**
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**
Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue
QString
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue?
```

**Related Commands**

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOURceport:VALue
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue
```

**Arguments**

<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

**Examples**

```
"XXXXXXXXXXXXXXXXXXXXXXXX000100010001" would set the sequence number that is to be used in the trigger to hexadecimal XXXX111.

```

**TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOURceport:VALue**

When the Ethernet trigger condition is set to TCPHeader, this command specifies the 16–bit source port address that is to be used in the trigger (along with the destination port address, the sequence number and the acknowledgement number). The default is all X’s (don’t care). The bus number is specified by <x>.

**Conditions**
Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

**Group**
Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOURceport:VALue
QString
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOURceport:VALue?
```
Related Commands

- TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition
- TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALUE
- TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALUE
- TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALUE

Arguments

<QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:TCPHeader:SOURceport:VALUE "XXXX000010100110" would set the source port address that is to be used in the trigger to hexadecimal X0A6.


TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition

This command specifies the condition to use when triggering on a FlexRay bus signal (start of frame, frame type, ID, cycle count, header, data, ID and data, EOF, error). The bus number is specified by <x>.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition
{SOF|FRAMEtype|IDENTifier|CYCLEcount|HEADER|DATA|IDANDDATA|EOF|ERROR}

TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition?

Arguments

SOF sets the trigger condition to start of frame.
FRAMEtype sets the trigger condition to frame type.
IDENTifier sets the trigger condition to identifier.
CYCLEcount sets the trigger condition to cycle count.
HEADER sets the trigger condition to header.
DATa sets the trigger condition to data.
IDANDDATA sets the trigger condition to id and data.
EOF sets the trigger condition to end of frame.
ERROR sets the trigger condition to error.

Examples


TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:HIVALue

This command specifies the high value when triggering on a FlexRay bus cycle count field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray CYCLEcount:VALue to set the low value.) The trigger condition must be set to CYCLEcount (using TRIGger:{A|B}:BUS:B<x>:FLEXray CONDITION). The bus number is specified by <x>.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:HIVALue?

Arguments <QString> is a quoted string that is the cycle count high value.

Examples TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:HIVALUE "110010" sets the cycle count high value to 110010.
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:QUALifier

This command specifies the qualifier (<, >, =, <=, >=, not =, in range, out of range) to use when triggering on the FlexRay bus cycle count field. The trigger condition must be set to CYCLEcount (using TRIGger:(A|B):BUS:B<x>:FLEXray:CONDition). The bus number is specified by <x>.

Conditions
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:QUALifier
{LESSthan|MOREthan|EQUAL|UNEQUAL|LESSEQUAL|MOREEQUAL|INrange|OUTrange}

TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:QUALifier?

Arguments
LESSthan sets the cycle count qualifier to less than.
MOREthan sets the cycle count qualifier to more than.
EQUAL sets the cycle count qualifier to equal.
UNEQUAL sets the cycle count qualifier to not equal.
LESSEQUAL sets the cycle count qualifier to less than or equal.
MOREEQUAL sets the cycle count qualifier to greater than or equal.
INrange sets the cycle count qualifier to in range.
OUTrange sets the cycle count qualifier to out of range.

Examples
TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:QUALIFIER LESSTHAN sets the cycle count qualifier to LESSTHAN.


TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:QUALIFIER EQUAL indicating that the cycle count qualifier is set to EQUAL.

TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:VALue

This command specifies the low value when triggering on the FlexRay bus cycle count field. (Use TRIGger:(A|B):BUS:B<x>:FLEXray:CYCLEcount:HIVALue to set the upper value.) The trigger condition must be set to CYCLEcount (using
Commands listed in alphabetical order

**TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition**. The bus number is specified by `<x>`.

**Conditions** Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group** Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:VALue?
```

**Arguments** `<QString>` is a quoted binary data string that represents the cycle count low value.

**Examples**

```
```

**TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:HIVALue**

This command specifies the high value when triggering on the FlexRay bus data field. (Use **TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:VALueto set the lower value.) The trigger condition needs to be set to ID or IDANDDATA (using **TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition**). The bus number is specified by `<x>`.

**Conditions** Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group** Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:HIVALue?
```

**Arguments** `<QString>` is a quoted string that is the binary data high value.
Examples

```
"11001101XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
xxxxx1"
```

sets the binary data string high value to

```
"11001101XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
xxxxx1"
```

```
```
might return

```
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXX" indicating the binary data string high value is don't care.

**TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:OFFSet**

This command specifies the offset of the data string, in bytes, when triggering on the FlexRay bus data field. The trigger condition needs to be set to ID or IDANDDATA (using **TRIGger:{A|B}:BUS:B<x>:FLEXray:CONdi tion**). The bus number is specified by <x>.

**Conditions**

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**

Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:OFFSet <NR1>
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:OFFSet?
```

**Arguments**

<NR1> is the offset of the data string in bytes. A byte offset of -1 signifies “don’t care”, and no byte offset is used. The instrument will trigger on or match any byte value that fits.

**Examples**

```
TRIGGER:A:BUS:B1:FLEXRAY:DATA:OFFSET 0 indicating that a data offset of 0.
```

**TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:QUALi fier**

This command specifies the qualifier (<, >, =, <=, >=, not =, in range, out of range) to use when triggering on the FlexRay bus data field. The trigger condition needs to be set to ID or IDANDDATA (using **TRIGger:{A|B}:BUS:B<x>:FLEXray:CONdi tion**). The bus number is specified by <x>. 
Commands listed in alphabetical order

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Trigger

**Syntax**
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:QUALifier
{LESSthan|MOREthan|EQual|UNEQual|LESSEQual|MOREEQual|INrange|OUTrange}

TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:QUALifier?

**Arguments**
LESSthan sets the data qualifier to less than.
MOREthan sets the data qualifier to greater than.
EEqual sets the data qualifier to equal.
UNEQual sets the data qualifier to not equal.
LESSEQual sets the data qualifier to less than or equal.
MOREEQual sets the data qualifier to greater than or equal.
INrange sets the data qualifier to in range.
OUTrange sets the data qualifier to out of range.

**Examples**
TRIGGER:A:BUS:B1:FLEXRAY:DATA:QUALIFIER LESSTHAN sets the data qualifier to LESSTHAN.

**TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:SIZE**

This command specifies the length of the data string, in bytes, when triggering on the FlexRay bus data field. The trigger condition needs to be set to ID or IDANDDATA (using TRIGger:(A|B):BUS:B<x>:FLEXray:CONDition). The bus number is specified by <x>.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Trigger
Syntax

TRIGGER:{A|B}:BUS:B<x>:FLEXray:DATA:SIZE <NR1>
TRIGGER:{A|B}:BUS:B<x>:FLEXray:DATA:SIZE?

Arguments

<NR1> is the FlexRay data string length, in bytes.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:DATA:SIZE 8 sets the data string size to 8 bytes.

TRIGGER:A:BUS:B1:FLEXRAY:DATA:SIZE 1 indicating the data size is 1 byte.

TRIGGER:{A|B}:BUS:B<x>:FLEXray:DATA:VALUE

This command specifies the low value when triggering on the FlexRay bus data field. (Use TRIGGER:{A|B}:BUS:B<x>:FLEXray:DATA:HIGHue to set the upper value.) The trigger condition needs to be set to ID or IDANDDATA (using TRIGGER:{A|B}:BUS:B<x>:FLEXray:CONDITION). The bus number is specified by <x>.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Trigger

Syntax

TRIGGER:{A|B}:BUS:B<x>:FLEXray:DATA:VALUE <QString>
TRIGGER:{A|B}:BUS:B<x>:FLEXray:DATA:VALUE?

Arguments

<QString> is a quoted string.

Examples


TRIGGER:A:BUS:B1:FLEXray:DATA:VALUE "XXXXXXXX" indicating the FlexRay data value is don't care.

TRIGGER:{A|B}:BUS:B<x>:FLEXray:EOFTYPE

This command specifies the end of file type (static, dynamic or any) when triggering on the FlexRay bus EOF field. The trigger condition needs to be set
to EOF (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by <x>.

Conditions
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:FLEXray:EOFTYPE {STATic|DYNAMic|ANY}
TRIGger:{A|B}:BUS:B<x>:FLEXray:EOFTYPE?

Arguments
STATic specifies triggering on the STATIC end of file type.
DYNAMic specifies triggering on the DYNAMIC end of file type.
ANY specifies triggering on a STATIC or DYNAMIC end of file type.

Examples
TRIGGER:A:BUS:B1:FLEXRAY:EOFTYPE ANY sets the FlexRay end of file type to ANY.
TRIGGER:A:BUS:B1:FLEXRAY:EOFTYPE STATIC indicating the FlexRay end of file type is STATIC

TRIGger:{A|B}:BUS:B<x>:FLEXray:ERRTYPE

This command specifies the error type when triggering on the FlexRay bus signal. The trigger condition needs to be set to ERROR (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by <x>.

Conditions
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:FLEXray:ERRTYPE {CRCHeader|CRCTrailer|SYNCFrame|STARTupnosync|NULLFRStatic|NULLFRDynamic}
TRIGger:{A|B}:BUS:B<x>:FLEXray:ERRTYPE?
Commands listed in alphabetical order

Arguments

- CRCHeader sets the error type to CRCHeader.
- CRCTrailer sets the error type to CRCTrailer.
- SYNFramen sets the error type to SYNFrame.
- STARTupnosync sets the error type to STARTupnosync.
- NULLFRStatic sets the error type to NULLFRStatic.
- NULLFRDynamic sets the error type to NULLFRDynamic.

Examples

```
Examples

TRIGGER:A:BUS:B1:FLEXRAY:ERRTYPE SYNCFRAME sets the trigger type
is SYNCFRAME.

TRIGGER:A:BUS:B1:FLEXRAY:ERRTYPE CRCHEADER indicating the
FlexRay trigger type is CRCHeader.
```

**TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:HIVALue**

This command specifies the high value when triggering on the FlexRay bus
frame ID field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:VALUE
to set the low value.) The trigger condition needs to be set to IDentifier (using
TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDITION). The bus number is specified
by <x>.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis
application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:HIVALue?
```

Arguments

- <QString> is a quoted string that is the binary frame ID high value.

Examples

```
Examples

TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:HIVALUE "11001100101" sets the
frame ID high value to 11001100101.

indicating the frame ID high value is "don't care".
```
**TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:QUALifier**

This command specifies the qualifier to use when triggering on the FlexRay bus frame ID field. The trigger condition needs to be set to IDentifier (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDITION). The bus number is specified by <x>.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Trigger

**Syntax**
```
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:QUALifier {LESSthan|MOREthan|EQual|UNEQual|LESSEQual|MORREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:QUALifier?
```

**Arguments**
- LESSthan sets the frame ID qualifier to less than.
- MOREthan sets the frame ID qualifier to greater than.
- EQual sets the frame ID qualifier to equal.
- UNEQual sets the frame ID qualifier to not equal.
- LESSEQual sets the frame ID qualifier to less than or equal.
- MORREEQual sets the frame ID qualifier to greater than or equal.
- INrange sets the frame ID qualifier to in range.
- OUTrange sets the frame ID qualifier to out of range.

**Examples**
```
```

**TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:VALue**

This command specifies the low value when triggering on the FlexRay bus frame ID field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:HIVALue to set the high value.) The trigger condition needs to be set to IDentifier (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDITION). The bus number is specified by <x>. 

---

2-1020 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
Commands listed in alphabetical order

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:VALUE <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:VALUE?
```

**Arguments**

`<QString>` is a quoted string that is the FlexRay frame ID low value.

**Examples**

```
```

**TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEType**

This command specifies the frame type (normal, payload, null, sync or startup) when triggering on the FlexRay bus signal. The trigger condition needs to be set to FRAMEType (using `TRIGger (A|B) BUS B<x> FLEXray:CONDITION`). The bus number is specified by `<x>`.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEType
{NORMAL|PAYLOAD|NULL|SYNC|STARTup}
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEType?
```

**Arguments**

`NORMAL` specifies the normal frame type.
`PAYLOAD` specifies the payload frame type.
`NULL` specifies the null frame type.
`SYNC` specifies the sync frame type.
`STARTup` specifies the startup frame type.
Examples

TRIGGER:A:BUS:B1:FLEXRAY:FRAMETYPE PAYLOAD sets the frame type to payload.

TRIGGER:A:BUS:B1:FLEXRAY:FRAMETYPE NORMAL indicating the frame type is set to normal.

TRIGGER:{A|B}:BUS:B<x>:FLEXRAY:HEADER:CRC

This command specifies the CRC portion of the binary header string when triggering on the FlexRay bus signal. The trigger condition needs to be set to HEADER (using TRIGGER:{A|B}:BUS:B<x>:FLEXRAY:CONDITION). The bus number is specified by <x>.

Conditions
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group
Trigger

Syntax
TRIGGER:{A|B}:BUS:B<x>:FLEXRAY:HEADER:CRC <QString>
TRIGGER:{A|B}:BUS:B<x>:FLEXRAY:HEADER:CRC?

Arguments
<QString> is a quoted string that is the CRC portion of the binary header string.

Examples
TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CRC "11001100101" sets the CRC portion of the binary header string to 11001100101.

TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CRC "XXXXXXXXXXX" indicating the CRC portion of the binary header string is don't care.

TRIGGER:{A|B}:BUS:B<x>:FLEXRAY:HEADER:CYCLECOUNT

This command specifies the cycle count portion of the binary header string when triggering on the FlexRay bus header. The trigger condition needs to be set to HEADER (using TRIGGER:{A|B}:BUS:B<x>:FLEXRAY:CONDITION). The bus number is specified by <x>.

Conditions
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.
Commands listed in alphabetical order

**Group**
Trigger

**Syntax**
- TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CYCLEcount <QString>
- TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CYCLEcount?

**Arguments**
- <QString> is a quoted string that is the cycle count portion of the binary header string.

**Examples**

**TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:FRAMEID**

This command specifies the frame ID portion of the binary header string when triggering on the FlexRay bus header. The trigger condition needs to be set to HEADer (using TRIGger:{A|B} BUS B <x> :FLEXray CONDITION). The bus number is specified by <x>.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Trigger

**Syntax**
- TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:FRAMEID <QString>
- TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:FRAMEID?

**Arguments**
- <QString> is a quoted string that represents the frame ID portion of the binary header string.

**Examples**
- TRIGGER:A:BUS:B1:FLEXRAY:HEADER:FRAMEID "11001100101" sets the frame ID portion of the binary header string to 11001100101.
**TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:INDBits**

This command specifies the indicator bits portion of the binary header string when triggering on the FlexRay bus header. The trigger condition needs to be set to HEADer (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by <x>.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Trigger

**Syntax**
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:INDBits <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:INDBits?

**Arguments**
QString is a quoted string that is the indicator bits portion of the binary header string.

**Examples**
TRIGGER:A:BUS:B1:FLEXRAY:HEADER:INDBITS "11001" sets the indicator bits portion of the header string to 11001.

**TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:PAYLength**

This command specifies the payload length portion of the binary header string when triggering on the FlexRay bus header. The trigger condition needs to be set to HEADer (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by <x>.

**Conditions**
Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**
Trigger

**Syntax**
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:PAYLength <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:PAYLength?
Arguments  

<Qstring> is the length of the payload portion of the Binary header string.

Examples


TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:MODE

This command specifies the I²C address mode to 7 or 10-bit. The bus number is specified by <x>.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group

Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:I2C:ADDRESS:MODE {ADDR7|ADDR10}

TRIGger:{A|B}:BUS:B<x>:I2C:ADDRESS:MODE?

Arguments

ADDR7 specifies the 7-bit I²C address mode.

ADDR10 specifies the 10-bit I²C address mode.

Examples


TRIGger:{A|B}:BUS:B<x>:I2C:ADDRESS:VALue

This command specifies the binary address string used for the I²C trigger if the trigger condition is ADDRESS or ADDRANDDATA. The bus number is specified by <x>.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.
 Commands listed in alphabetical order

**Group** Trigger

**Syntax**

*TRIGger:*{A|B}:BUS:B<x>:I2C:ADDRESS:VALUE <QString>

*TRIGger:*{A|B}:BUS:B<x>:I2C:ADDRESS:VALUE?

**Arguments**

<QString> is up to 7 or 10-bits depending on the address mode that specifies the address. The only allowed characters in the QString are 0, 1, and X.

**Examples**

TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE "1011" sets the I2C address value to XXX1011.


:TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE "XXXXXXX" indicating the address value is set to XXXXXXX.

---

**TRIGger:{A|B}:BUS:B<x>:I2C:CONDition**

This command specifies the trigger condition for an I2C trigger. The bus number is specified by <x>.

**Conditions**

Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

**Group** Trigger

**Syntax**

*TRIGger:*{A|B}:BUS:B<x>:I2C:CONDITION

{START|STOP|REPEATstart|ACKMISS|ADDRESS|DATa|ADDRANDDATA}

*TRIGger:*{A|B}:BUS:B<x>:I2C:CONDITION?

**Arguments**

START specifies a search based on start condition.

STOP specifies a search based on stop condition.

REPEATstart specifies a search based on repeat of start condition.

ACKMISS specifies a search based on missing acknowledgement condition.

ADDRESS specifies a search based on address.

DATa specifies a search based on data.

ADDRANDDATA specifies a search based on address and data.
Examples


TRIGger:{A|B}:BUS:B<x>:I2C:DATa:DIRection

This command specifies the I2C trigger type to be valid on a Read, Write, or Either condition. Read or write is indicated by the R/W bit in the I2C protocol. The bus number is specified by <x>.

Conditions
Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group
Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:I2C:DATa:DIRection
{READ|WRITE|NOCARE}

TRIGger:{A|B}:BUS:B<x>:I2C:DATa:DIRection?

Arguments

READ specifies read as the data direction.

WRITE specifies write as the data direction.

NOCARE specifies either as the data direction.

Examples


TRIGger:{A|B}:BUS:B<x>:I2C:DATa:SIZe

This command specifies the length of the data string in bytes to be used for an I2C trigger if the trigger condition is DATA or ADDRANDDATA. Applies to bus <x>, where the bus number is specified by <x>.

Conditions
Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.
Commands listed in alphabetical order

**Group**  
**Trigger**

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:I2C:DATA:SIZE <NR1>
TRIGger:{A|B}:BUS:B<x>:I2C:DATA:SIZE?
```

**Arguments**

<NR1> is the length of the data string in bytes.

**Examples**

```
TRIGger:A:BUS:B1:I2C:DATA:SIZE 1 sets the data size to 1 byte.
:TRIGGER:A:BUS:B1:I2C:DATA:SIZE 1 indicating the size is set
to 1 byte.
```

**TRIGger:{A|B}:BUS:B<x>:I2C:DATa:VALue**

This command specifies the binary data string used for I2C triggering if the trigger condition is DATA or ADDRANDDATA. The bus number is specified by <x>.

**Conditions**

Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

**Group**  
**Trigger**

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:I2C:DATA:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:I2C:DATA:VALue?
```

**Arguments**

<QString> is the binary data string, where the number of bits is 8 times the number of bytes specified. The only allowed characters in the string are 0, 1, and X.

**Examples**

```
TRIGger:A:BUS:B1:I2C:DATA:VALue "11001101" sets the data value to 1100101.
:TRIGGER:A:BUS:B1:I2C:DATA:VALUE "XXXXXXXX" indicating the data value is XXXXXXXX.
```
TRIgger:{A|B}:BUS:B<x>:LIN:CONDition

This command specifies the trigger condition for LIN. The bus number is specified by <x>.

**Conditions**: Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**: Trigger

**Syntax**

TRIgger:{A|B}:BUS:B<x>:LIN:CONDition

{SYNCfield|IDentifier|DATa|IDANDDATA|WAKEup|SLEEP|ERRor}

TRIgger:{A|B}:BUS:B<x>:LIN:CONDition?

**Arguments**

SYNCfield sets the LIN trigger condition to sync field.

IDENTifier sets the LIN trigger condition to identifier.

DATa sets the LIN trigger condition to data.

IDANDDATA sets the LIN trigger condition to id and data.

WAKEup sets the LIN trigger condition to wake up.

SLEEP sets the LIN trigger condition to sleep.

ERROR sets the LIN trigger condition to error.

**Examples**


TRIgger:{A|B}:BUS:B<x>:LIN:DATa:HIValue

This command specifies the high data value string used for a LIN bus trigger when the trigger condition is DATA or IDANDDATA and the data qualifier is INRANGE or OUTRANGE. The bus number is specified by <x>.

**Conditions**: Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.
Commands listed in alphabetical order

**Group** Trigger

**Syntax**

- TRIGger:{A|B}:BUS:B<x>:LIN:DATa:HIVALue <QString>
- TRIGger:{A|B}:BUS:B<x>:LIN:DATa:HIVALue?

**Arguments**

<QString> is a quoted string that is the binary data string used for LIN trigger if the trigger condition is ID or IDANDDATA.

**Examples**

- TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE "11001010" sets the high value to 11001010.

**TRIGger:{A|B}:BUS:B<x>:LIN:DATa:QUALifier**

This command specifies the LIN data qualifier. This only applies if the trigger condition is IDANDDATA or DATA. The bus number is specified by <x>.

**Conditions** Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group** Trigger

**Syntax**

- TRIGger:{A|B}:BUS:B<x>:LIN:DATa:QUALifier {LESSthan|MOREthan|EQual|UNEQual|LESSEQual|MORREEQual|INrange|OUTrange}
- TRIGger:{A|B}:BUS:B<x>:LIN:DATa:QUALifier?

**Arguments**

- LESSthan sets the LIN data qualifier to less than.
- MOREthan sets the LIN data qualifier to greater than.
- EQual sets the LIN data qualifier to equal.
- UNEQual sets the LIN data qualifier to not equal.
- LESSEQual sets the LIN data qualifier to less than or equal.
- MORREEQual sets the LIN data qualifier to greater than or equal.
- INrange sets the LIN data qualifier to in range.
OUTrange sets the LIN data qualifier to out of range.

**Examples**

```
```

**TRIGger:{A|B}:BUS:B<x>:LIN:DATa:SIZe**

This command specifies the length of the data string in bytes to be used for LIN trigger. The bus number is specified by `<x>`.

**Conditions**

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**

Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:SIZe?
```

**Arguments**

`<NR1>` is the size of the data string in bytes.

**Examples**

```
TRIGGER:A:BUS:B1:LIN:DATA:SIZE 8 sets the data size to 8 bytes.
```

**TRIGger:{A|B}:BUS:B<x>:LIN:DATa:VALue**

This command specifies the binary data string to be used for LIN trigger condition if trigger condition is ID or IDANDDATA. The bus number is specified by `<x>`.

**Conditions**

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**

Trigger
Commands listed in alphabetical order

**Syntax**

TRIGger:{A|B}:BUS:B<x>:LIN:DATA:VALUE <QString>
TRIGger:{A|B}:BUS:B<x>:LIN:DATA:VALUE?

**Arguments**

<QString> is a quoted string that is the LIN trigger data value.

**Examples**

TRIGGER:A:BUS:B1:LIN:DATA:VALUE "11001101" sets the data value to 11001101.

TRIGGER:A:BUS:B1:LIN:DATA:VALUE "XXXXXXXX" indicating the data value is don't care.

**TRIGger:{A|B}:BUS:B<x>:LIN:ERRTYPE**

This command specifies the error type be used for LIN trigger. The bus number is specified by <x>.

**Conditions**

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

**Group**

Trigger

**Syntax**

TRIGger:{A|B}:BUS:B<x>:LIN:ERRTYPE {SYNC|PARity|CHECKsum}
TRIGger:{A|B}:BUS:B<x>:LIN:ERRTYPE?

**Arguments**

SYNC sets the LIN error type to SYNC.

PARity sets the LIN error type to parity.

CHECKsum sets the LIN error type to checksum.

**Examples**

TRIGGER:A:BUS:B1:LIN:ERRTYPE CHECKSUM sets the LIN error type to checksum.

TRIGGER:A:BUS:B1:LIN:ERRTYPE SYNC indicating the LIN error type is SYNC.

**TRIGger:{A|B}:BUS:B<x>:LIN:IDENTifier:VALue**

This command specifies the binary address string used for LIN bus trigger if the trigger condition is ID or IDANDDATA. The bus number is specified by <x>.
<table>
<thead>
<tr>
<th>Conditions</th>
<th>Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Trigger</td>
</tr>
<tr>
<td>Syntax</td>
<td>TRIGger:{A</td>
</tr>
<tr>
<td></td>
<td>TRIGger:{A</td>
</tr>
<tr>
<td>Arguments</td>
<td>&lt;QString&gt; is the binary address string used for LIN trigger if the trigger condition is ID or IDANDDATA.</td>
</tr>
<tr>
<td></td>
<td>TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE &quot;XXXXXX&quot; indicating the identifier value is XXXXXX.</td>
</tr>
</tbody>
</table>

**TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRESS:HIVALue**

This command sets or queries the high value when triggering on command word addresses for a MIL-STD-1553 bus. The bus number is specified by $x$. The trigger condition must be set to COMMAND, and the address qualifier must be INrange or OUTrange.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Trigger</td>
</tr>
<tr>
<td>Syntax</td>
<td>TRIGger:{A</td>
</tr>
<tr>
<td></td>
<td>TRIGger:{A</td>
</tr>
<tr>
<td>Arguments</td>
<td>&lt;QString&gt; is the address value.</td>
</tr>
<tr>
<td></td>
<td>TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:HIVALUE? might return &quot;XXXXXX&quot;, indicating that the value is XXXXXX.</td>
</tr>
</tbody>
</table>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:QUALifier

This command sets or queries the qualifier to be used when triggering on command word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:QUALifier {EQUAL|UNEQUAL|LESSthan|MOREthan
|LESSEqual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:QUALifier?

Arguments
Arguments are the available address qualifiers.

Examples
TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRess:QUALIFIER? might return EQUAL, indicating that the address qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:VALue

This command sets or queries the low value when triggering on command word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:VALue
<QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:VALue?

Arguments
<QString> is the address value.
Examples


TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:VALUE? might return "XXXXX", indicating that the value is XXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:COUNt

This command sets or queries the value of the command word "word count" field for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:COUNt <QString>

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:COUNt?

Arguments

<QString> is the word count value.

Examples

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:COUNT "X1000" sets the value to X1000.

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:COUNT? might return "XXXXX", indicating that the value is XXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:PARity

This command sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger
Commands listed in alphabetical order

**Syntax**

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:PARity
  {ONE|ZERO|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:PARity?

**Arguments**

ONE filters command words to only match those where the parity bit has a value of 1.

ZERO filters command words to only match those where the parity bit has a value of 0.

NOCARE disables filtering of command words on the parity bit.

**Examples**

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:PARITY ONE specifies filtering command words for those where the parity bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:PARITY? might return NOCARE, indicating that command words are not being filtered based on the parity bit value.

**TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:SUBADdress**

This command sets or queries the value of the command word subaddress field for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to COMMAND.

**Conditions**

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**

Trigger

**Syntax**

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:SUBADdress <QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:SUBADdress?

**Arguments**

<QString> is the subaddress value.

**Examples**

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:SUBADDRESS "X1000" sets the value to X1000.

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:SUBADDRESS? might return "XXXXXX", indicating that the value is XXXXX.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:TRBit

This command sets or queries the value of the command word Transmit / Receive bit for a MIL-STD-1553 bus to trigger on. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:TRBit {RX|TX|X}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:TRBit?

Arguments RX filters command words to only match those that are receive packets.
TX filters command words to only match those that are transmit packets.
X disables filtering of command words on the R/T bit.

Examples TRIGGER:A:BUS:B1:MIL1553B:COMMAND:TRBIT TX specifies filtering command words for only transmit messages.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:CONDition

This command sets or queries the field or condition for a MIL-STD-1553 bus to trigger on. The bus number is specified by x.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax TRIGger:{A|B}:BUS:B<x>:MIL1553B:CONDition {SYNC|COMMAND|STATus|DATA |TIME|ERROR}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:CONDition?
Arguments

SYNC specifies triggering on the sync pulse of any word.
COMMAND specifies triggering on a matching command word.
STATUS specifies triggering on a matching status word.
DATA specifies triggering on a matching data word.
TIME specifies triggering on the response time or intermessage gap between words.
ERROR specifies triggering on a specified error condition.

Examples

TRIGGER:A:BUS:B1:MIL1553B:CONDITION DATA specifies finding matching data word(s).
TRIGGER:A:BUS:B1:MIL1553B:CONDITION? might return SYNC, indicating that the bus is triggering on sync pulses found in any word.

TRIGGER:{A|B}:BUS:B<x>:MIL1553B:DATa:PARity

This command sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to DATA.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Trigger

Syntax
TRIGGER:{A|B}:BUS:B<x>:MIL1553B:DATa:PARity
{ONE|ZERO|NOCARE}
TRIGGER:{A|B}:BUS:B<x>:MIL1553B:DATa:PARity?

Arguments
ONE filters data words to only match those where the parity bit has a value of 1.
ZERO filters data words to only match those where the parity bit has a value of 0.
NOCARE disables filtering of data words on the parity bit.

Examples
TRIGGER:A:BUS:B1:MIL1553B:DATa:PARITY ONE specifies filtering data words for those where the parity bit has the value 1.
TRIGGER:A:BUS:B1:MIL1553B:DATa:PARITY? might return NOCARE, indicating that data words are not being filtered based on the parity bit value.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:VALue

This command sets or queries the value when triggering on data words for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to DATA.

**Conditions** Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group** Trigger

**Syntax**

TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:VALue?

**Arguments**<QString> is the data value.

**Examples**

TRIGGER:A:BUS:B1:MIL1553B:DATA:VALUE "XXXXXXXXXXXX1000" sets the value to XXXXXXXXXXXX1000.

TRIGGER:A:BUS:B1:MIL1553B:DATA:VALUE? might return "XXXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:ERRTYPe

This command sets or queries the type of error condition for a MIL-STD-1553 bus to trigger on. The bus number is specified by x. The trigger condition must be set to ERRor.

**Conditions** Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group** Trigger

**Syntax**

TRIGger:{A|B}:BUS:B<x>:MIL1553B:ERRTYPe {PARity|SYNC|DATA}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:ERRTYPe?

**Arguments**

PARity specifies triggering on an incorrectly calculated parity bit in any word.

SYNC specifies triggering on any sync pulse that does not transition in the middle of the pulse as required.
DATA specifies triggering on any non-contiguous data words.

**Examples**


TRIGGER:A:BUS:B1:MIL1553B:ERRTYPE? might return PARITY, indicating that the bus is triggering on parity errors in any word.

**TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:HIVALue**

This command sets or queries the high value when triggering on status word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to STATus and the address qualifier must be INrange or OUTrange.

**Conditions**

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**

Trigger

**Syntax**

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:HIVALue

 TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:HIVALue?

**Arguments**

<QString> is the address value.

**Examples**


TRIGGER:A:BUS:B1:MIL1553B:STATus:ADDRess:HIVALue? might return "XXXXX", indicating that the value is XXXXX.

**TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:QUALifier**

This command sets or queries the qualifier to be used when triggering on status word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to STATus.

**Conditions**

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.
Commands listed in alphabetical order

Group Trigger

Syntax TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:QUALifier {Equal|UNEQual|LESSthan |MOREthan|LESSEqual|MOREEQual|INrange|OUTrange} TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:QUALifier?

Arguments Arguments are the available address qualifiers.


TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:QUALIFIER? might return EQUAL, indicating that the address qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:VALue

This command sets or queries the low value when triggering on status word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger


Arguments <QString> is the address value.


TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:VALUE? might return "XXXXX", indicating that the value is XXXXX.
Commands listed in alphabetical order

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BCR

This command sets or queries the value of the broadcast command received bit (BCR bit, bit 15) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BCR
{ONE|ZERO|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BCR?

Arguments
ONE filters status words to only match those where the BCR bit has a value of 1.
ZERO filters status words to only match those where the BCR bit has a value of 0.
NOCARE disables filtering of status words on the BCR bit.

Examples
TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:BCR? might return NOCARE, indicating that status words are not being filtered based on the BCR bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BUSY

This command sets or queries the value of the busy bit (BUSY bit, bit 16) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BUSY
{ONE|ZERO|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BUSY?
### Arguments
- **ONE** filters status words to only match those where the BUSY bit has a value of 1.
- **ZERO** filters status words to only match those where the BUSY bit has a value of 0.
- **NOCARE** disables filtering of status words on the BUSY bit.

### Examples
- ```plaintext
```
specifies filtering status words for those where the BUSY bit has a value of 1.
- ```plaintext
```
indicating that status words are not being filtered based on the BUSY bit value.

### TRIGger:\{A|B\}:BUS:B<x>:MIL1553B:STATus:BIT:DBCA

This command sets or queries the value of the dynamic bus control acceptance bit (DBCA bit, bit 18) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

### Conditions
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

### Group
Trigger

### Syntax
```plaintext
TRIGGER:{A|B}:BUS:B<x>:MIL1553B:STATUS:BIT:DBCA
{ONE|ZERO|NOCARE}
TRIGGER:{A|B}:BUS:B<x>:MIL1553B:STATUS:BIT:DBCA?
```

### Arguments
- **ONE** filters status words to only match those where the DBCA bit has a value of 1.
- **ZERO** filters status words to only match those where the DBCA bit has a value of 0.
- **NOCARE** disables filtering of status words on the DBCA bit.

### Examples
- ```plaintext
```
specifies filtering status words for those where the DBCA bit has a value of 1.
- ```plaintext
```
indicating that status words are not being filtered based on the DBCA bit value.

### TRIGger:\{A|B\}:BUS:B<x>:MIL1553B:STATus:BIT:INSTR

This command sets or queries the value of the instrumentation bit (INSTR bit, bit 10) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.
Commands listed in alphabetical order

**Conditions**
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**
Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:INSTR
ONE|ZERO|NOCARE
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:INSTR?
```

**Arguments**

ONE filters status words to only match those where the INSTR bit has a value of 1.

ZERO filters status words to only match those where the INSTR bit has a value of 0.

NOCARE disables filtering of status words on the INSTR bit.

**Examples**

```

TRIGGER:A:BUS:B1:MIL1553B:STATus:BIT:INSTR? might return NOCARE, indicating that status words are not being filtered based on the INSTR bit value.
```

**TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:ME**

This command sets or queries the value of the message error bit (ME bit, bit 9) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

**Conditions**
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**
Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:ME
ONE|ZERO|NOCARE
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:ME?
```

**Arguments**

ONE filters status words to only match those where the ME bit has a value of 1.

ZERO filters status words to only match those where the ME bit has a value of 0.

NOCARE disables filtering of status words on the ME bit.
**TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:ME**

Examples

<table>
<thead>
<tr>
<th>Specifications</th>
<th>Examples</th>
</tr>
</thead>
</table>

**TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SRQ**

This command sets or queries the value of the status word service request bit (SRQ bit, bit 11) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

**Syntax**

TRIGGER:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SRQ

{ONE|ZERO|NOCARE}

TRIGGER:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SRQ?

**Arguments**

ONE filters status words to only match those where the SRQ bit has a value of 1.

ZERO filters status words to only match those where the SRQ bit has a value of 0.

NOCARE disables filtering of status words on the SRQ bit.

**TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SUBSF**

This command sets or queries the value of the subsystem flag bit (SUBSF bit, bit 17) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

**Conditions**

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.
### TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATUS:BIT:SUBSF

<table>
<thead>
<tr>
<th>Group</th>
<th>Trigger</th>
</tr>
</thead>
</table>
| Syntax    | TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATUS:BIT:SUBSF
            | {ONE|ZERO|NOCARE}       |
|           | TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATUS:BIT:SUBSF? |
| Arguments | ONE filters status words to only match those where the SUBSF bit has a value of 1.
            | ZERO filters status words to only match those where the SUBSF bit has a value of 0.
            | NOCARE disables filtering of status words on the SUBSF bit. |

### TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATUS:BIT:TF

<table>
<thead>
<tr>
<th>Group</th>
<th>Trigger</th>
</tr>
</thead>
</table>
| Syntax    | TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATUS:BIT:TF
            | {ONE|ZERO|NOCARE}       |
|           | TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATUS:BIT:TF? |
| Arguments | ONE filters status words to only match those where the TF bit has a value of 1.
            | ZERO filters status words to only match those where the TF bit has a value of 0.
            | NOCARE disables filtering of status words on the TF bit. |
TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:TF? might return NOCARE, indicating that status words are not being filtered based on the TF bit value.

TRIGGER:{A|B}:BUS:B<x>:MIL1553B:STATUS:PARity

This command sets or queries the value of the status word parity bit for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATUS.

**Conditions**
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**
Trigger

**Syntax**
TRIGGER:{A|B}:BUS:B<x>:MIL1553B:STATUS:PARity
{ONE|ZERO|NOCARE}
TRIGGER:{A|B}:BUS:B<x>:MIL1553B:STATUS:PARity?

**Arguments**
- **ONE** filters status words to only match those where the parity bit has a value of 1.
- **ZERO** filters status words to only match those where the parity bit has a value of 0.
- **NOCARE** disables filtering of status words on the parity bit.

**Examples**
- TRIGGER:A:BUS:B1:MIL1553B:STATUS:PARITY? might return NOCARE, indicating that status words are not being filtered based on the parity bit value.

TRIGGER:{A|B}:BUS:B<x>:MIL1553B:TIME:LESSLimit

This command sets or queries the lower limit to be used when triggering on response time / inter message gap time for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to TIME.

**Conditions**
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**
Trigger
Commands listed in alphabetical order

**Syntax**

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIME:LESSLimit <NR3>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIME:LESSLimit?

**Related Commands**

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIME:QUALifier

**Arguments**

<NR3> is a floating point number that specifies the lower bound for measuring remote terminal response time (RT) or the inter-message gap (IMG) in seconds.

**Examples**

TRIGGER:A:BUS:B1:MIL1553B:TIME:LESSLIMIT 2.0000e-6 sets the lower bound for comparison to 2 microseconds.

TRIGGER:A:BUS:B1:MIL1553B:TIME:LESSLIMIT? might return 4.0000E-6, indicating that the lower bound for comparison is set to 4 microseconds.

**TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIME:MORELimit**

This command sets or queries the upper limit to be used when triggering on response time / inter message gap time for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to TIME.

**Conditions**

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**

Trigger

**Syntax**

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIME:MORELimit <NR3>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIME:MORELimit?

**Related Commands**

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIME:QUALifier

**Arguments**

<NR3> is a floating point number that specifies the upper bound for measuring remote terminal response time (RT) or the inter-message gap (IMG) in seconds.

**Examples**

TRIGGER:A:BUS:B1:MIL1553B:TIME:MORELIMIT 15.0000e-6 sets the upper bound for comparison to 15 microseconds.

TRIGGER:A:BUS:B1:MIL1553B:TIME:MORELIMIT? might return 12.0000E-6, indicating that the upper bound for comparison is set to 12 microseconds.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIME:QUALifier

This command sets or queries the qualifier to be used when triggering on response time / inter message gap time for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to TIME.

**Conditions**
Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

**Group**
Trigger

**Syntax**
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIME:QUALifier {LESSthan|MOREthan|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIME:QUALifier?

**Related Commands**
TRIGger:(A|B) BUS:B<x>:MIL1553B TIME:LESSLimit
TRIGger:(A|B) BUS:B<x>:MIL1553B TIME:MORELimit

**Arguments**
Arguments are the available comparison qualifiers.

**Examples**
TRIGGER:A:BUS:B1:MIL1553B:TIME:QUALIFIER LESSTHAN sets the time comparison qualifier to less than.

TRIGger:{A|B}:BUS:B<x>:PARallel:DATA:VALue

This command specifies the binary data string used for a Parallel Bus trigger. The bus number is specified by <x>.

**Group**
Trigger

**Syntax**
TRIGger:{A|B}:BUS:B<x>:PARallel:DATA:VALUE <QString>
TRIGger:{A|B}:BUS:B<x>:PARallel:DATA:VALUE?

**Arguments**
QString is the binary data string used for a Parallel Bus trigger.
Commands listed in alphabetical order

Examples

TRIGger:A:BUS:B1:PARallel:DATA:VALUE "11001101" sets the data value to 11001101.


TRIGger:{A|B}:BUS:B<x>:RS232C:CONDition

This command specifies the condition for an RS-232C trigger, where the bus number is specified by <x>.

Conditions
Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:RS232C:CONDition
{START|EOp|DATA|PARityerror}  
TRIGger:{A|B}:BUS:B<x>:RS232C:CONDition?

Arguments
START sets the Trigger on condition to Start.
EOp sets the Trigger on condition to End of Packet.
DATA sets the Trigger on condition to Data.
PARityerror sets the Trigger on condition to Parity Error.

Examples
TRIGger:A:BUS:B1:RS232C:CONDition DATA sets the trigger on condition to data.


TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:SIZe

This command sets or queries the length of the data string in bytes to be used for an RS-232C trigger when the trigger condition is Data. The bus number is specified by <x>. 

2-1050  MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
Commands listed in alphabetical order

**Conditions**
Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

**Group**
Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:SIZe <NR3>
```

```
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:SIZe?
```

**Arguments**
<NR3> specifies the data size in bytes.

**Examples**

```
```
sets three bytes data size for the RS-232C bus B1 trigger.

```
:TRIGGER:A:BUS:B1:RS232C:DATA:SIZE 2, indicating that the
data size for the RS-232C bus B1 trigger is set to two bytes.
```

**TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:VALue**

This command sets or queries the data address string used for the RS-232 bus trigger when the trigger condition is set to Data. The bus number is specified by <x>.

**Conditions**
Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

**Group**
Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:VALue <QString>
```

```
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:VALue?
```

**Arguments**
<QString> specifies the address value. The argument is a string of 0, 1, or X representing a binary number.

**Examples**

```
```
sets the data address string used for the RS-232 bus trigger to 011XX11.

```
that the data address string used for the RS-232 bus trigger is set to "XXXXXXX01"
**TRIGger:{A|B}:BUS:B<x>:SENT:CONDition**

This command sets or queries the trigger condition for a SENT bus.

**Conditions**
Requires option SRAUTOSEN.

**Group**
Trigger

**Syntax**
- TRIGger:{A|B}:BUS:B<x>:SENT:CONDITION
  - {START|FAST|SLOW|ERRor}
- TRIGger:{A|B}:BUS:B<x>:SENT:CONDITION?

**Arguments**
- **B<x>** is the number of the bus waveform.
- **START** sets triggering on start of packet.
- **FAST** sets triggering on fast channel packets.
- **SLOW** sets triggering on slow channel packets.
- **ERRor** sets triggering on errors.

**Examples**

**TRIGger:{A|B}:BUS:B<x>:SENT:ERRType**

This command sets or queries the error type to be used when triggering on SENT data.

**Conditions**
Requires option S5SRAUTOSEN.

The SENT bus trigger condition is set to ERRor.

**Group**
Trigger

**Syntax**
- TRIGger:{A|B}:BUS:B<x>:SENT:ERRType_CRC
- TRIGger:{A|B}:BUS:B<x>:SENT:ERRType?
### TRIGger\{A|B\}:BUS:B<x>:SENT:CONDition

**Arguments**
- B<x> is the number of the bus waveform.
- CRC specifies triggering on CRC errors.

**Examples**

### TRIGger\{A|B\}:BUS:B<x>:SENT:ERRType:CRC

**This command sets or queries the CRC error type to be used when triggering on SENT data.**

**Conditions**
- Requires option SRAUTOSEN.
- The SENT bus trigger condition is set to ERRor.
- The ERRType is set to CRC.

**Group**
- Trigger

**Syntax**
- TRIGger\{A|B\}:BUS:B<x>:SENT:ERRType:CRC {FAST|SLOW}
- TRIGger\{A|B\}:BUS:B<x>:SENT:ERRType:CRC?

**Related Commands**
- TRIGger\{A|B\}:BUS:B<x>:SENT:CONDition
- TRIGger\{A|B\}:BUS:B<x>:SENT:ERRType

**Arguments**
- B<x> is the number of the bus waveform.
- FAST specifies triggering on CRC errors in only the Fast Channel.
- SLOW specifies triggering on CRC errors in only the slow channel.

**Examples**
- TRIGGER:A:BUS:B1:SENT:ERRType:CRC FAST to indicate that the bus is triggering on Fast Channel CRC errors.
**TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:HIVALue**

This command sets or queries the high binary fast channel 1 value to use when triggering on a SENT bus signal.

**Conditions**
Requires option SRAUTOSEN.
The SENT bus trigger condition must be set to FAST.

**Group**
Trigger

**Syntax**
```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:HIVALue?
```

**Related Commands**
TRIGger (A|B):BUS B<x>:SENT CONDITION

**Arguments**
- `Bus<x>` is the bus identifier number.
- `<QString>` sets the Fast Channel 1 binary data high value.

**Examples**
```
TRIGger:A:BUS:B12:SENT:FAST:CHAN1A:HIVALue "XXXXXXXXXXXX" sets the SENT bus B12 Fast Channel 1 high value to XXXXXXXXXXXX, or "don't care."
```
```
```

**TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier**

This command sets or queries the qualifier to be used when triggering on SENT fast packet bus data for device channel 1.

**Conditions**
Requires option SRAUTOSEN
The SENT bus trigger condition must be set to FAST.

**Group**
Trigger
Commands listed in alphabetical order

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier {EQual|UNEQual|LESSthan|MOREthan|
LESSEQual|MORREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier?
```

Related Commands

```
TRIGger:(A|B):BUS:B<x>:SENT:CONDITION
TRIGger:(A|B):BUS:B<x>:SENT:FAST:CHAN1A:VALUE
```

Arguments

- `B<x>` is the number of the bus.
- `EQUal` sets the qualifier as Equal.
- `LESSEQual` sets the qualifier as Less Than or Equal to.
- `LESSThan` sets the qualifier as Less Than.
- `MOREEQual` sets the qualifier as More Than or Equal to.
- `MOREThan` sets the qualifier as More Than.
- `UNEQual` sets the qualifier as Unequal.
- `INrange` sets the qualifier to inside a range.
- `OUTrange` sets the qualifier to outside a range.

Examples

```
TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:QUALIFIER MOREEQUAL to indicate that the fast channel 1 data qualifier trigger is set to greater than or equal for bus 5.
```

**TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:VALUE**

This command sets or queries the binary fast channel 1 value to be used when triggering on a SENT bus signal.

Conditions

- Requires option SRAUTOSEN.

The search trigger condition must be set to FAST.

Group

Trigger
Syntax
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:VALue?

Related Commands
TRIGger:{A|B}:BUS:B<x>:SENT:CONDition
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier

Arguments
B<x> is the number of the bus waveform.
Qstring> is the Fast Channel 1 value on which to trigger.

Examples
TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:VALUE "XXXXXXXXXXXX" sets the
Fast Channel 1 value to trigger on XXXXXXXXXXXX, or "don't care."
:TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:VALUE "0000XXXX1111" to
indicate the Fast Channel 1 binary trigger value is 0000XXXX1111.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:HIVALue
This command sets or queries the high binary fast channel 2 value to use when
triggering on a SENT bus signal.

Conditions
Requires option SRAUTOSEN.
The SENT bus trigger condition must be set to FAST.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:HIVALue?

Related Commands
TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

Arguments
Bus<x> is the bus identifier number.
Qstring> sets the Fast Channel 2 high binary data value.

Examples
the SENT bus B2 Fast Channel 2 high value to 100000000000.
indicating the SENT bus B4 Fast Channel 2 high is set to the binary value
0101xxxx1111.

TRIGGER:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:QUALifier

This command sets or queries the qualifier to be used when triggering on SENT
fast packet bus data for device channel 2.

Conditions
Requires option SRAUTOSEN.
The SENT bus trigger condition must be set to FAST.

Group
Trigger

Syntax
TRIGGER:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:QUALifier {EQUAL|UNEQUAL|LESS THAN|MORE THAN|
LESSEQUAL|MOREEQUAL|IN RANGE|OUT RANGE}
TRIGGER:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:QUALifier?

Related Commands
TRIGGER:{A|B}:BUS:B<x>:SENT:CONDITION
TRIGGER:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:VALUE

Arguments
B<x> is the number of the bus.
EQUAL sets the qualifier as Equal.
LESSEQUAL sets the qualifier as Less Than or Equal to.
LESSTHAN sets the qualifier as Less Than.
MOREEQUAL sets the qualifier as More Than or Equal to.
MORETHAN sets the qualifier as More Than.
UNEQUAL sets the qualifier as Unequal.
IN RANGE sets the qualifier to inside a range.
OUT RANGE sets the qualifier to outside a range.

Examples
channel 2 data qualifier trigger to not equal for bus 2.
indicate that the fast channel 2 data qualifier trigger is set to greater than or equal
for bus 5.

TRIGGER:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:VALUE

This command sets or queries the binary fast channel 2 value to be used when
triggering on a SENT bus signal. The trigger condition must be set to FAST.

Conditions
Requires option SRAUTOSEN.
The search condition must be set to FAST.

Group
Trigger

Syntax
TRIGGER:A|B}:BUS:B<x>:SENT:FAST:CHAN2B:VALUE <Qstring>
TRIGGER:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:VALUE?

Related Commands
TRIGGER:{A|B}:BUS:B<x>:SENT:CONDition
TRIGGER:{A|B}:BUS:B<x>:SENT:PAUSE:QUALifier

Arguments
B<x> is the number of the bus waveform.
<Qstring> is the Fast Channel 2 binary value on which to trigger.

Examples
TRIGGER:A:BUS:B1:SENT:FAST:CHAN2B:VALUE "111111111111" sets the
Fast Channel 2 value on which to trigger to 111111111111.

indicate the Fast Channel 2 binary value on which to trigger is 000000000000.

TRIGGER:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:HIVALue

This command sets or queries the high binary fast message counter value to be
used when triggering on a SENT bus signal.

Conditions
Requires option SRAUTOSEN.
The SENT bus trigger condition must be set to FAST.
The number of channels must be set to 1.
The nibble count must be set to 6.

Group Trigger

Syntax TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:HIVALue <QString>
       TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:HIVALue?

Related Commands TRIGger (A|B): BUS:B<x>: SENT:CONDITION
                 BUS:B<x>: SENT:NUMCHANnel
                 BUS:B<x>: SENT:NIBBLECount

Arguments Bus<x> is the bus identifier number.
            <QString> sets the Fast Channel 1 counter binary value.

Examples TRIGGER:A:BUS:B1:SENT:FAST:COUNTER:HIVALUE "XXXXXXXX" sets the
         Fast Channel 1 secure counter high value to "don't care."
         TRIGGER:A:BUS:B3:SENT:FAST:COUNTER:HIVALUE "11110000" to
         indicate a binary value of 11110000.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:QUALifier

This command sets or queries the qualifier to be used when triggering on SENT
fast packet bus data for the secure format counter.

Conditions Requires option SRAUTOSEN.
               The SENT bus trigger condition must be set to FAST.

Group Trigger

Syntax TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:QUALifier {Equal|UNEQual|LESSthan|MOREthan|
               LESSEEQual|MORREEQual|INrange|OUTrange}
       TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:QUALifier?

Related Commands TRIGger (A|B): BUS:B<x>: SENT:CONDITION
TRIGGER: (A|B) : BUS:B<x>:SENT:FAST:COUNTer:VALue

Arguments
B<x> is the number of the bus.
EQUAL sets the qualifier as Equal.
LESSEQUAL sets the qualifier as Less Than or Equal to.
LESThan sets the qualifier as Less Than.
MOREEQual sets the qualifier as More Than or Equal to.
MOREThan sets the qualifier as More Than.
UNEQual sets the qualifier as Unequal.
INrange sets the qualifier to inside a range.
OUTrange sets the qualifier to outside a range.

Examples

TRIGger: {A|B}:BUS:B<x>:SENT:FAST:COUNTer:VALue

This command sets or queries the binary fast message counter value to be used when triggering on a SENT bus signal.

Conditions
Requires option SRAUTOSEN.
The search condition must be set to FAST.
The number of channels must be set to 1.
The nibble count must be set to 6.

Group
Trigger

Syntax
TRIGger: {A|B}:BUS:B<x>:SENT:FAST:COUNTer:VALue <Qstring>
TRIGger: {A|B}:BUS:B<x>:SENT:FAST:COUNTer:VALue?

Related Commands
TRIGger: (A|B) : BUS:B<x>:SENT:CONDITION
**Arguments**

B<x> is the number of the bus waveform.

<Qstring> is the Fast Channel 1 fast message counter binary value on which to trigger.

**Examples**

TRIGGER:A:BUS:B1:SENT:FAST:COUNTER:VALUE "XXXXXXXX" sets the Fast Channel 1 fast message counter value to "don't care."


**TRIGGER:{A|B}:BUS:B<x>:SENT:FAST:INVERTNIBBLE:VALUE**

This command sets or queries the binary fast message inverted nibble value to be used when triggering on a SENT bus signal.

**Conditions**

Requires option SRAUTOSEN.

The search condition must be set to FAST.

The number of channels must be set to 1.

The nibble count must be set to 6.

**Group**

Trigger

**Syntax**

TRIGGER:{A|B}:BUS:B<x>:SENT:FAST:INVERTNIBBLE:VALUE

<Qstring>

TRIGGER:{A|B}:BUS:B<x>:SENT:FAST:INVERTNIBBLE:VALUE?

**Related Commands**

TRIGGER:{A|B}:BUS:B<x>:SENT:CONDITION

BUS:B<x>:SENT:NIBBLECount

BUS:B<x>:SENT:NUMCHANNEL

**Arguments**

B<x> is the number of the bus waveform.

<Qstring> is the Fast Channel 1 inverted nibble binary value on which to trigger.
Commands listed in alphabetical order

**Examples**

TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE "XXXX" sets the Fast Channel 1 secure inverted nibble value to "don't care."


**TRIGger:{A|B}:BUS:B<x>:SENT:FAST:STATus:VALue**

This command sets or queries the binary status value to be used when triggering on a SENT bus signal.

**Conditions**
Requires option SRAUTOSEN.
The search trigger condition must be set to FAST.

**Group**
Trigger

**Syntax**

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:STATus:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:STATus:VALue?

**Related Commands**
TRIGger:(A|B).BUS:B<x>:SENT:CONDITION

**Arguments**

B<x> is the number of the bus waveform.

<Qstring> is the binary status value on which to trigger.

**Examples**

TRIGGER:A:BUS:B1:SENT:FAST:STATus:VALue "XXXX" sets the binary status trigger value to XXXX.


**TRIGger:{A|B}:BUS:B<x>:SENT:PAUSE:QUALifier**

This command sets or queries the qualifier to be used when triggering on SENT pause pulses.

**Conditions**
Requires option SRAUTOSEN.
The SENT bus trigger condition must be set to PAUSE.
Group  Trigger

Syntax  TRIGger:{A|B}:BUS:B<x>:SENT:PAUSE:QUALifier {EQUAL|UNEQUAL|
        LESSThan|MOREThan|LESSEEQual|MOREEQual|INrange|OUTrange}
        TRIGger:{A|B}:BUS:B<x>:SENT:PAUSE:QUALifier?

Related Commands  TRIGger:{A|B}:BUS:B<x>:SENT:CONDITION

Arguments  B<x> is the bus identifier number.
        EQUAL sets the qualifier as Equal.
        INrange sets the qualifier to in range.
        LESSEEQual sets the qualifier as Less Than or Equal to.
        LESSThan sets the qualifier as Less Than.
        MOREEQual sets the qualifier as More Than or Equal to.
        MOREThan sets the qualifier as More Than.
        OUTrange sets the qualifier to out of range.
        UNEQUAL sets the qualifier as Unequal.

        qualifier trigger to not equal for bus 2.
        :TRIGGER:A:BUS:B1:SENT:PAUSE:QUALIFIER INRANGE to indicate that
        the pause pulse qualifier is set to trigger inside the range of ticks low and ticks
        high on bus 1.

TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:HIVALue

This command sets or queries the high binary Slow channel data value to use
when triggering on a SENT bus signal.

Conditions  Requires option SRAUTOSEN.
        The SENT bus trigger condition must be set to SLOW.
Group: Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:HIVALue?

Related Commands: TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

Arguments

Bus<x> is the bus identifier number.

<QString> sets the binary Slow channel data value.

Examples

TRIGGER:A:BUS:B1:SENT:SLOW:DATA:HIVALUE "XXXXXXXX" sets the Slow data high value to "don't care."

TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:QUALifier

This command sets or queries the binary identifier value to use when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN

The SENT bus trigger condition must be set to SLOW.

Group: Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:QUALifier {EQUAL|UNEQUAL|LESSthan|MOREthan|LESSEqual|MOREEQual}
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:QUALifier?

Related Commands: TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

Arguments

B<x> is the bus identifier number.

EQUAL specifies the qualifier as Equal.
UNEQUAL specifies the qualifier as Not Equal to.
LESSthan specifies the qualifier as Less Than.
MOREthan specifies the qualifier as More Than.
LESSEqual specifies the qualifier as Less Than or Equal to.
MOREEQUAL specifies the qualifier as More Than or Equal to.
INrange sets the qualifier to inside a range.
OUTrange sets the qualifier to outside a range.

Examples

TRIGGER:{A|B}:BUS:B2:SENT:SLOW:DATA:QUALIFIER LESSEQUAL sets the slow channel data qualifier to less than or equal for bus 2.


TRIGGER:{A|B}:BUS:B<x>:SENT:SLOW:DATA:VALUE

This command sets or queries the binary slow channel value to use when triggering on a SENT bus signal.

Conditions
Requires option SRAUTOSEN
The SENT bus trigger condition must be set to SLOW.

Group
Trigger

Syntax
TRIGGER:{A|B}:BUS:B<x>:SENT:SLOW:DATA:VALUE <Qstring>
TRIGGER:{A|B}:BUS:B<x>:SENT:SLOW:DATA:VALUE?

Related Commands
TRIGGER:{A|B}:BUS:B<x>:SENT:CONDITION

Arguments
B<x> is the bus identifier number.
<Qstring> is the binary slow channel data value.

Examples


TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:IDENTifier:VALUE

This command sets or queries the qualifier to use when triggering on SENT slow packet bus data.

Conditions
Requires option SRAUTOSEN
The SENT bus trigger condition must be set to SLOW.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:IDENTifier:VALUE <Qstring>
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:IDENTifier:VALUE

Related Commands
TRIGger:{A|B}:BUS:B<x>:SENT:CONDITION
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:VALUE

Arguments
B<x> is the bus identifier number.
<Qstring> is the binary identifier value.

Examples

TRIGger:{A|B}:BUS:B<x>:SPI:CONDITION

This command specifies the trigger condition for a SPI trigger. The bus number is specified by <x>.

Conditions
Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:SPI:CONDITION {SS|STARTofframe|DATA}
TRIGger:{A|B}:BUS:B<x>:SPI:CONDITION?
Related Commands

- BUS:B<x>:SPI:IDLETime
- BUS:B<x>:SPI:FRAMING

Arguments

B<x> is the bus identifier number.

SS specifies the Slave Selection condition.

STARTofframe is applicable when BUS:B<x>:SPI:FRAMING is set to IDLEtime. When the trigger condition is set to STARTofframe, the instrument triggers on the first SPI clock after an idle time when there are no clocks.

DATA sets the trigger condition to Master-In Slave-Out and Master-Out Slave-In.

TRIGGER:{A|B}:BUS:B<x>:SPI:DATa:SIZe

This command specifies the length of the data string to be used for a SPI trigger if the trigger condition is DATA. The bus number is specified by <x>.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Trigger

Syntax

TRIGger:{A|B}:BUS:B<x>:SPI:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:SIZe?

Arguments

B<x> is the bus identifier number.

<NR1> is the length of the data string in bytes.

Examples

TRIGger:A:BUS:B1:SPI:DATa:SIZe 1 sets the data size to 1 byte.

TRIGger:{A|B}:BUS:B<x>:SPI:DATa:VALue

This command specifies the binary data string used for SPI triggering if the trigger condition is DATA. The bus number is specified by <x>. 
Commands listed in alphabetical order

**Conditions**
Requires option 5-SR.COMP or SUP5-SR.COMP Triggering and Analysis application.

**Group**
Trigger

**Syntax**
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:VALue?

**Arguments**

\( B<x> \) is the bus identifier number.

\(<QString>\) specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format.

**Examples**
TRIGger:A:BUS:B1:SPI:DATa:VALue "11011010" sets the data value to 11011010.

**TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition**

This command sets or queries the trigger condition for the specified SPMI bus.

**Conditions**
Requires option SRPM

**Group**
Trigger

**Syntax**
TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition {SSC|RESet|SLEep|SHUTdown|Wakeup|MASTErREAD|MASTErWRITE|REGeator|REGWRITE|DEVIceDESCSLAVERead|DEVIceDESCSLAVEReAd|EXTREGREAD|EXTREGWRITE|LONGEXTREGREAD|LONGEXTREGWRITE|REG0WRITE|AUTHenticate|TRANSferbusownership|PARityerror}
TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition?

**Arguments**

\( B<x> \) is the bus identifier number.

SSC specifies triggering on the Sequence Start Condition.

RESet specifies triggering on the Reset command sequence.

SLEep specifies triggering on the Sleep command sequence.
SHUTDOWN specifies triggering on the Shutdown command sequence.
WAKEUP specifies triggering on the Wakeup command sequence.
MASTERREAD specifies triggering on the Master Read command sequence.
MASTERWRITE specifies triggering on the Master Write command sequence.
REGREAD specifies triggering on the Register Read command sequence.
REGWRITE specifies triggering on the Register Write command sequence.
DEVICEDESCMASTERREAD specifies triggering on the Device Descriptor Block Master Read command sequence.
DEVICEDESCSLAVEREAD specifies triggering on the Device Descriptor Block Slave Read command sequence.
EXTREGREAD specifies triggering on the Extended Register Read command sequence.
EXTREGWRITE specifies triggering on the Extended Register Write command sequence.
LONGEXTREGREAD specifies triggering on the Extended Register Read Long command sequence.
LONGEXTREGWRITE specifies triggering on the Extended Register Write Long command sequence.
REG0WRITE specifies triggering on the Register 0 Write command sequence.
AUTHENTICATE specifies triggering on the Authentication command sequence.
TRANSFERBUSOWNERSHIP specifies triggering on the Transfer Bus Ownership (TBO) command sequence.
PARITYERROR specifies triggering on the parity errors.

**Examples**


**TRIGGER:{A|B}:BUS:B<x>:SPMI:DATA:SIZE**

This command sets or queries the length of the data string, in bytes, to be used when triggering on an SPMI bus signal.

**Conditions**

Requires option SRPM.
The trigger condition must be set to EXTREGREAD, EXTREGWRITE, LONGEXTREGREAD, or LONGEXTREGWRITE.

Note that this a floating point value. If the condition is set to REG0WRITE, the query return value will be a fractional value (0.875) because the data size is less than a byte.

**Group** Trigger

**Syntax**

```plaintext
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:SIZe <NR2>
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:SIZe?
```

**Related Commands**

- TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition
- TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:VALue

**Arguments**

- `Bus<x>` is the bus identifier number.
- `<NR2>` is the size of the data string in bytes.

**Examples**

```
```

sets the data value to 1 byte.

```
```

might return

```
```

to indicate the data pattern has four bytes of data.

**TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:VALue**

This command sets or queries the binary data string used for SPMI triggering if the trigger condition is MASTERREAD, MASTERWRITE, REGREAD, REGWRITE, EXTREGREAD, EXTREGWRITE, LONGEXTREGREAD, LONGEXTREGWRITE, or REG0WRITE.

**Conditions**

- Requires option SRPM

**Group** Trigger

**Syntax**

```plaintext
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:VALue?
```

**Related Commands**

- TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition
Arguments

*Bus<x>* is the bus identifier number.

*<Qstring>* is the binary data string that identifies the data value on which to trigger.

Examples

TRIGGER:A:BUS:B1:SPMI:DATA:VALUE "XXXXXXXX" sets the binary data value on which to trigger for bus B1 to "don't care."


**TRIGger:{A|B}:BUS:B<x>:SPMI:MASTERADDRess:VALue**

This command sets or queries the binary data string that identifies the master address used in SPMI triggering if the trigger condition is MASTERREAD, MASTERWRITE, or DEVICEDESCMASTERREAD.

Arguments

*Bus<x>* is the bus identifier number.

*<Qstring>* is the binary data string that identifies the master address on which to trigger.

Examples

TRIGGER:A:BUS:B3:SPMI:MASTERADDRESS:VALUE "XX" sets the binary master address trigger of bus B3 to "don't care."


**TRIGger:{A|B}:BUS:B<x>:SPMI:NORESPonse**

This command sets or queries whether or not to trigger on No Response frames.
### TRIGger:{A|B}:BUS:B<x>:SPMI:NORESPonse \(<Qstring>\)

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:SPMI:NORESPonse <Qstring>
TRIGger:{A|B}:BUS:B<x>:SPMI:NORESPonse?
```

**Conditions**

Requires option SRPM

**Group**

Trigger

**Related Commands**

TRIGger\(\{A|B\}\):BUS B \(x\):SPMI:CONDition

**Arguments**

- \(Bus<x>\) is the bus identifier number.
- \(<Qstring>\) is either TRUE or FALSE boolean value.

**Examples**


### TRIGger:{A|B}:BUS:B<x>:SPMI:REGISTERADDRess:VALue

This command sets or queries the binary data string that identifies the register address used in SPMI triggering if the trigger condition is MASTERREAD, MASTERWRITE, REGREAD, REGWRITE, EXTREGREAD, EXTREGWRITE, LONGEXTREGREAD, or LONGEXTREGWRITE.

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:SPMI:REGISTERADDRess:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SPMI:REGISTERADDRess:VALue?
```

**Conditions**

Requires option SRPM

**Group**

Trigger

**Related Commands**

TRIGger\(\{A|B\}\):BUS B \(x\):SPMI:CONDition

**Arguments**

- \(bus<x>\) is the bus identifier number.
- \(<Qstring>\) is the binary data string that identifies the register address.
Examples


TRIGger:{A|B}:BUS:B<x>:SPMI:SLAVEADDRESS:VALue

This command sets or queries the binary data string that identifies the slave address used in SPMI triggering if the trigger condition is RESet, SLEEP, SHUTDOWN, WAKEup, AUTHenticate, REGREAD, REGWRite, EXTREGREAD, EXTREGWRite, LONGEXTREGREAD, LONGEXTREGWRite, DEVICEDESCSLAVEREAd, or REG0WRite.

Conditions
Requires option SRPM

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:SPMI:SLAVEADDRESS:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SPMI:SLAVEADDRESS:VALue?

Related Commands
TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition

Arguments
Bus<x> is the bus identifier number.

<Qstring> is the binary data string that identifies the slave address on which to trigger.

Examples


TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:HIVALue

This command specifies the binary address string for the upper limit for inside-of-range and outside-of-range qualifiers for the USB trigger. Use the command TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:VALue to set the lower limit. The bus number is specified by <x>. 
Commands listed in alphabetical order

**Conditions** Requires option 5-SRUSB 2 or SUP5-SRUSB 2 Triggering and Analysis application.

**Group** Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:USB:ADDRESS:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:ADDRESS:HIVALue?
```

**Related Commands**

```
TRIGger:{A|B}:BUS:B<x>:USB:CONDITION
TRIGger:{A|B}:BUS:B<x>:USB:ADDRESS:VALUE
```

**Arguments**

- `B<x>` is the bus identifier number.
- `<QString>` within the range 0000000 to 1111111 (00 hex to 7F hex).

**Examples**

```
TRIGGER:A:BUS:B1:USB:ADDRESS:HIVALUE "0001000" sets the upper limit to binary 0001000 (08 hex).
that the upper limit is 1111111 (7F hex).
```

**TRIGger:{A|B}:BUS:B<x>:USB:ADDRESS:VALUE**

This command specifies the binary address string to be used for USB trigger. The trigger condition must be set to TOKEN. The bus number is specified by `<x>`.

**Conditions** Requires option 5-SRUSB 2 or SUP5-SRUSB 2 Triggering and Analysis application.

**Group** Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:USB:ADDRESS:VALUE <QString>
TRIGger:{A|B}:BUS:B<x>:USB:ADDRESS:VALUE?
```

**Related Commands**

```
TRIGger:{A|B}:BUS:B<x>:USB:CONDITION
```
Arguments  

B<x> is the bus identifier number.  

<QString> within the range 0000000 to 1111111 (00 hex to 7F hex).

Examples  

TRIGGER:A:BUS:B1:USB:ADDRESS:VALUE "0001000" sets the binary address to 0001000 (08 hex).


:TRIGGER:A:BUS:B1:USB:ADDRESS:VALUE "1000000", which indicates that the binary address is 100000 (40 hex).

TRIGger:{A|B}:BUS:B<x>:USB:CONDition

This command specifies the trigger condition for the USB trigger. The bus number is specified by <x>.

Conditions  

Requires option 5-SRUSB 2 or SUP5-SRUSB 2 Triggering and Analysis application.

Group  

Trigger

Syntax  

TRIGGER:{A|B}:BUS:B<x>:USB:CONDITION  
{SYNC|RESET|SUSPEND|RESUME|EOP|TOKENPacket|DATAPacket|  
HANDSHAKEPacket|SPECIALPacket|ERROR} 

TRIGger:{A|B}:BUS:B<x>:USB:CONDITION?

Arguments  

SYNC indicates triggering on a Sync field of a packet.  

RESET sets triggering on a reset condition.  

SUSPEND sets triggering on a suspend condition.  

RESUME sets triggering on a resume condition.  

EOP indicates triggering on an end-of-packet signal.  

TOKENPacket indicates triggering on a token packet.  

DATAPacket indicates triggering on a data packet.  

HANDSHAKEPacket indicates triggering on a handshake packet.  

SPECIALPacket indicates triggering on a special status packet.  

ERROR indicates triggering on an error condition.
Examples

TRIGGER:A:BUS:B1:USB:CONDITION TOKENPACKET sets the trigger condition to be a token packet.


TRIGger:{A|B}:BUS:B<x>:USB:DATa:HIVALue

This command specifies the binary data string for the upper limit for inside-of-range and outside-of-range qualifiers for the USB trigger when the trigger condition is DATA. Use the command TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue to set the lower limit. The bus number is specified by <x>.

Conditions
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:USB:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:DATa:HIVALue?

Related Commands
TRIGger:{A|B}:BUS:B<x>:USB:CONDition
TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue

Arguments
<QString> within the range 00000000 to 11111111 (00 hex to FF hex).

Examples


TRIGger:{A|B}:BUS:B<x>:USB:DATa:OFFSet

This command specifies the data offset in bytes to trigger on. The bus number is specified by <x>.

Conditions
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.
Commands listed in alphabetical order

**Group** Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:OFFSet <NR1>
TRIGger:{A|B}:BUS:B<x>:USB:DATa:OFFSet?
```

**Related Commands** TRIGger:{A|B}:BUS:B<x>:USB:CONDition

**Arguments**

<NR1> is number in the range 0 to 1024.

**Examples**

```
:TRIGGER:A:BUS:B1:USB:DATA:OFFSET 0, indicating that the
data offset is the default value, 0.
```

**TRIGger:{A|B}:BUS:B<x>:USB:DATa:QUALifier**

This command sets the qualifier to be used when triggering on a USB bus signal. The trigger condition must be set to DATAPACKET. The bus number is specified by <x>.

**Conditions** Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

**Group** Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:QUALifier {EQUAL|UNEQual|LESSthan|MOREthan|LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:USB:DATa:QUALifier?
```

**Arguments**

EQUAL specifies the qualifier as Equal.

INrange specifies the qualifier as Inside Range.

LESSEqual specifies the qualifier as Less Than or Equal to.

MORREEQual specifies the qualifier as More Than or Equal to.

OUTrange specifies the qualifier as Out of Range.

UNEQual specifies the qualifier as Not Equal to.

LESSthan specifies the qualifier as Less Than.
MORE than specifies the qualifier as More Than.

Examples


TRIGGER:{A|B}:BUS:B<x>:USB:DATA:SIZE

This command specifies the number of contiguous data bytes to trigger on. The bus number is specified by <x>.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

TRIGGER:{A|B}:BUS:B<x>:USB:DATA:SIZE <NR1>
TRIGGER:{A|B}:BUS:B<x>:USB:DATA:SIZE?

Arguments

<NR1> is number in the range 1 to 16.

Examples


TRIGGER:{A|B}:BUS:B<x>:USB:DATA:TYPE

This command specifies the data type for a USB trigger. The bus number is specified by <x>.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger
Commands listed in alphabetical order

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:TYPE
{ANY|DATA0|DATA1|DATA2|MDATA}
TRIGger:{A|B}:BUS:B<x>:USB:DATa:TYPE?
```

**Related Commands**

```
TRIGger:{A|B}:BUS:B<x>:USB:CONDition
```

**Arguments**

- **ANY** indicates either a DATA0 or DATA1 data packet type.
- **DATA0** indicates a DATA0 data packet type.
- **DATA1** indicates a DATA1 data packet type.
- **DATA2** indicates a DATA2 data packet type when on HIGH speed.
- **MDATA** indicates a MDATA data packet type when on HIGH speed.

**Examples**

```
TRIGGER:A:BUS:B1:USB:DATA:TYPE DATA0 sets the oscilloscope to trigger on a DATA0 data packet type.
```

**TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue**

This command specifies the binary data string to be used when triggering on a USB trigger.

**Conditions**

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

**Group**

Trigger

**Syntax**

```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue?
```

**Related Commands**

```
TRIGger:{A|B}:BUS:B<x>:USB:CONDition
```

**Arguments**

- **<QString>** within the range 00000000 to 11111111 (00 hex to FF hex).

**Examples**

```
TRIGGER:A:BUS:B1:USB:DATA:VALUE "00001000" sets the binary address to 00001000 (08 hex).
```
Commands listed in alphabetical order

indicates that the binary address is 0100000 (40 hex).

TRIGGER:{A|B}:BUS:B<x>:USB:ENDPoint:VALUE

This command specifies the binary endpoint string to be used for the USB trigger. The bus number is specified by <x>.

Conditions Requires option 5-SRUSB 2 or SUP5-SRUSB 2 Triggering and Analysis application.

Group Trigger

Syntax TRIGGER:{A|B}:BUS:B<x>:USB:ENDPoint:VALUE <QString>
TRIGGER:{A|B}:BUS:B<x>:USB:ENDPoint:VALUE?

Arguments <QString> within the range 0000 to 1111 (00 hex to 0F hex).

Examples TRIGGER:A:BUS:B1:USB:ENDPOINT:VALUE "1000" sets the binary address to 1000 (08 hex).
:TRIGGER:A:BUS:B1:USB:ENDPOINT:VALUE "1001", which indicates that the binary address is 1001 (09 hex).

TRIGGER:{A|B}:BUS:B<x>:USB:ERRType

This command specifies the error type to be used when the trigger condition is set to ERROR. The bus number is specified by <x>.

Conditions Requires option 5-SRUSB 2 or SUP5-SRUSB 2 Triggering and Analysis application.

Group Trigger
Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:ERRType
{PID|CRC5|CRC16|BITSTUFFing}
TRIGger:{A|B}:BUS:B<x>:USB:ERRType?
```

Related Commands

```
TRIGger.(A|B).BUS.B <x>:USB:CONDition
```

Arguments

- **PID** indicates the error type is set to packet ID.
- **CRC5** indicates the error type is set to 5-bit CRC.
- **CRC16** indicates the error type is set to 16-bit CRC.
- **BITSTUFFing** indicates the error type is set to bit stuffing.

Examples

```
TRIGGER:A:BUS:B1:USB:ERRTYPE PID sets the error trigger condition to packet ID.
```

**TRIGger:{A|B}:BUS:B<x>:USB:HANDSHAKEType**

This command specifies the handshake type for the USB trigger. The bus number is specified by `<x>`.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:HANDSHAKEType
{ANY|NAK|ACK|STALL|NYET}
TRIGger:{A|B}:BUS:B<x>:USB:HANDSHAKEType?
```

Related Commands

```
TRIGger.(A|B).BUS.B <x>:USB:CONDition
```

Arguments

- **ANY** indicates the oscilloscope will trigger on any handshake type.
- **NAK** indicates the oscilloscope will trigger when a device cannot send or receive data.
- **ACK** indicates the oscilloscope will trigger when a packet is successfully received.
STALL indicates the oscilloscope will trigger when a device requires intervention from the host.

NYET specifies the handshake type as No response Yet (0110).

Examples

TRIGGER:A:BUS:B1:USB:HANDSHAKETYPE ACK sets the handshake type to acknowledgement.


**TRIGGER:{A|B}:BUS:B<x>:USB:SOFFRAMENUMber**

This command specifies the binary data string to be used for start of frame number, when the trigger condition is Token Packet and the token type is Start of Frame. The bus number is specified by `<x>`.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

TRIGGER:{A|B}:BUS:B<x>:USB:SOFFRAMENUMber <QString>

TRIGGER:{A|B}:BUS:B<x>:USB:SOFFRAMENUMber?

Related Commands

TRIGGER:{A|B}:BUS:B<x>:USB:CONDITION

Arguments

<QString> within the range 000 0000 0000 to 111 1111 1111 (000 hex to 7FF hex).

Examples

TRIGGER:A:BUS:B1:USB:SOFFRAMENUMBER "00000001000" sets the start of frame number to 00000001000 (008 hex).


**TRIGGER:{A|B}:BUS:B<x>:USB:SPECIALType**

This command specifies the packet ID (PID) for the special packet. The bus number is specified by `<x>`.
Conditions
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType
{ANY|ERROR|PING|PREAMble|RESERVED|SPLit}
TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType?

Related Commands
TRIGger:{A|B}:BUS:B<x>:USB:CONDition

Arguments
ANY specifies the PID value as Any (XX00).
ERROR specifies the PID value as ERR (1100).
PING specifies the PID value as PING (0100).
PREAMble specifies the PID value as PRE (1100).
RESERVED specifies the PID value as Reserved (0000).
SPLit specifies the PID value as Split (1000).

Examples
TRIGGER:A:BUS1:USB:SPECIALTYPE PREAMBLE sets the special packet type to preamble.
TRIGGER:A:BUS1:USB:SPECIALTYPE? might return
:TRIGGER:A:BUS1:USB:SPECIALTYPE PREAMBLE, indicating that the special type is set to preamble.

TRIGger:{A|B}:BUS:B<x>:USB:SPLit:ET:VALue

When triggering on a high-speed USB split transaction, this command specifies the split transaction endpoint type value to trigger on. The bus number is specified by <x>.

Conditions
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:ET:VALue
{NOCARE|CONTRol|ISOchronous|BULK|INTERRUPT}
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:ET:VALue?
Related Commands

**TRIGger\{(A|B)\}:BUS:B<x>:USB:SPECIALType**

Arguments

- NOCARE — any endpoint type.
- CONTROL — control endpoint type.
- ISOCherent — isochronous endpoint type.
- BULK — bulk endpoint type (BULK-IN or BULK-OUT).
- INTERRUPT — interrupt endpoint type (Interrupt-IN).

Examples

- `TRIGger:A:BUS:B1:USB:SPLit:ET:VALUE?` might return BULK, indicating that the bulk endpoint type has been specified to trigger on.

**TRIGger:\{A|B\}:BUS:B<x>:USB:SPLit:HUB:VALue**

When triggering on a high-speed USB split transaction, this command specifies the split transaction hub address value to trigger on. The trigger condition must be set to Special with packet type SPLIT. The value can be up to 7 characters long. The default is all X’s (don’t care). The bus number is specified by <x>.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

- `TRIGger:\{A|B\}:BUS:B<x>:USB:SPLit:HUB:VALUE <QString>`
- `TRIGger:\{A|B\}:BUS:B<x>:USB:SPLit:HUB:VALUE?`

Related Commands

- `TRIGger\{(A|B)\}:BUS:B<x>:USB:SPECIALType`

Arguments

- `QString` is a quoted string of up to 7 characters. The valid characters are 0 and 1.

Examples

- `TRIGger:A:BUS:B1:USB:SPLit:HUB:VALUE "001010"` sets the split transaction hub address to trigger on to 001010.
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:PORT:VALue

When triggering on a high-speed USB split transaction, this command specifies the split transaction port address value to trigger on. The trigger condition must be set to Special with a packet type SPLIT. The value can be up to 7 characters long. The default is all X’s (don’t care). The bus number is specified by <x>.

Conditions
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:PORT:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:PORT:VALue?

Related Commands
TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType

Arguments
QString is a quoted string of up to 7 characters. The valid characters are 0 and 1.

Examples
TRIGger:A:BUS:B1:USB:SPLit:PORT:VALue "001010" sets the split transaction hub address to trigger on to 001010.
TRIGger:A:BUS:B1:USB:SPLit:PORT:VALue? might return XXXXXXXX, indicating that the hub address value to trigger on doesn’t matter.

TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SC:VALue

When triggering on a high-speed USB split transaction, this command specifies whether to trigger on the start or complete phase of the split transaction, based on the Start/Complete bit field value. (0 = Start, 1 = Complete). The default is NOCARE. The bus number is specified by <x>.

Conditions
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SC:VALue {NOCARE|SSPLIT|CSPLIT}
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SC:VALue?
Related Commands  
TRIGger\{A\|B\}:BUS:B<x>:USB:SPECIALType

Arguments  
NOCARE — trigger on either the start or complete phase of the split transaction.
SSPLIT — trigger on the start phase of the split transaction.
CSPLIT — trigger on the complete phase of the split transaction.

Examples  
TRIGger:A:BUS:B1:USB:SPLIT:SC:VALUE? might indicate NOCARE, specifying that it doesn’t matter whether to trigger on the start or complete phase of the split transaction.

TRIGger:{A\|B}:BUS:B<x>:USB:SPLIT:SE:VALUE

When triggering on a high-speed USB split transaction, this command specifies the split transaction start/end bit value to trigger on. The bus number is specified by <x>.

**NOTE.** The start and end bits are interpreted based on the type of split transaction:

*For Interrupt and control transactions, the S bit means Speed: 0 = Full Speed, 1 = Low Speed.*

*For bulk IN/OUT and isochronous IN start-split transactions, the S field must be 0.*

*For bulk/control IN/OUT, interrupt IN/OUT, and isochronous IN start-split transactions, the E field must be 0.*

*For full-speed isochronous OUT start-split transactions, the S (Start) and E (End) fields specify how the high-speed data payload corresponds to data for a full-speed data packet as shown below:*  

<table>
<thead>
<tr>
<th>S</th>
<th>E</th>
<th>High-speed to Full-speed Data Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>High-speed data is the middle of the full-speed data payload.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>High-speed data is the end of the full-speed data payload.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>High-speed data is the beginning of the full-speed data payload.</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>High-speed data is all of the full speed data payload.</td>
</tr>
</tbody>
</table>

Conditions  
Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.
Group       Trigger

Syntax TRIGger:{A|B}:BUS:B<x>:USB:SPli:t:SE:VALue
{NOCARE|FULLSPEED|LOWSPEED|ISOSTART|ISOMID|ISOEND|ISOALL}
TRIGger:{A|B}:BUS:B<x>:USB:SPli:t:SE:VALue?

Related Commands TRIGger:(A|B):BUS:B<x>:USB:SPECIALType

Arguments NOCARE — any combination of S and E bits.
FULLSPEED — S bit = 0, E bit = 0.
LOWSPEED — S bit = 1, E bit = 0.
ISOSTART — S bit = 1, E bit = 0.
ISOMID — see note above.
ISOEND — see note above.
ISOALL — see note above.

the ISOEND split transaction value.

TRIGger:{A|B}:BUS:B<x>:USB:TOKENType

This command specifies the token type for the USB trigger. The bus number is
specified by <x>.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis
application.

Group       Trigger

Syntax TRIGger:{A|B}:BUS:B<x>:USB:TOKENType {ANY|SOF|OUT|IN|SETUP}
TRIGger:{A|B}:BUS:B<x>:USB:TOKENType?

Related Commands TRIGger:(A|B):BUS:B<x>:USB:CONDition
Arguments

- ANY indicates any of the token types.
- SOF indicates a SOF (start-of-frame) token type.
- OUT indicates an OUT token type.
- IN indicates an IN token type.
- SETUP indicates a SETUP token type.

Examples

- TRIGGER:A:BUS:B1:USB:TOKENTYPE SETUP sets the token type to SETUP.
  :TRIGGER:A:BUS:B1:USB:TOKENTYPE SOF if the token type is SOF.

TRIGger:{A|B}:BUS:SOURce

This command sets or queries the source bus for a bus trigger.

Group
Trigger

Syntax
TRIGger:{A|B}:BUS:SOURce B<x>
TRIGger:{A|B}:BUS:SOURce?

Arguments

- B<x> sets the selected source to the bus.

Examples


TRIGger:{A|B}:EDGE:COUPling

This command sets or queries the type of coupling for the edge trigger. This command is equivalent to selecting Edge from the Trigger Type drop-down in the Trigger setup context menu, and choosing from the Coupling drop-down list.

Group
Trigger

Syntax
TRIGger:{A|B}:EDGE:COUPling {DC|HFRej|LFRej|NOISErej}
TRIGger:{A|B}:EDGE:COUPling?
**Related Commands**

TRIGger\{(A|B)\}:EDGE:SOURce
 TRIGger\{(A|B)\}:EDGE:SLOpe

**Arguments**

DC selects DC trigger coupling, which passes all input signals to the trigger circuitry.

HFRej coupling attenuates signals above 50 kHz before passing the signals to the trigger circuitry.

LFRej coupling attenuates signals below 80 kHz before passing the signals to the trigger circuitry.

NOISERej coupling provides stable triggering by increasing the trigger hysteresis. Increased hysteresis reduces the trigger sensitivity to noise but can require greater trigger signal amplitude.

**Examples**

TRIGGER:A:EDGE:COUPLING DC sets the A edge trigger coupling to DC.

TRIGGER:A:EDGE:COUPLING? might return :TRIGGER:A:EDGE:COUPLING DC, indicating that the A edge trigger coupling is set to DC.

**TRIGger\{(A|B)\}:EDGE:SLOpe**

This command sets or queries the slope for the edge trigger. This command is equivalent to selecting Edge from the Trigger Type drop-down in the Trigger setup context menu, and then choosing the desired Slope. This command is also equivalent to pressing the front-panel Slope button.

**Group**

Trigger

**Syntax**

TRIGger\{(A|B)\}:EDGE:SLOpe \{RISe|FALL|EITHER\}

TRIGger\{(A|B)\}:EDGE:SLOpe?

**Related Commands**

TRIGger\{(A|B)\}:EDGE:SOURce
 TRIGger\{(A|B)\}:EDGE:COUPLing
 TRIGger:B:STATE

**Arguments**

RISe specifies to trigger on the rising or positive edge of a signal.

FALL specifies to trigger on the falling or negative edge of a signal.

EITHER specifies to trigger on either the rising or falling edge of a signal.
Examples

TRIGGER:A:EDGE:SLOPE RISE sets the A edge trigger slope to positive, which triggers on the rising edge of the signal.


TRIGger:{A|B}:EDGE:SOUrce

This command sets or queries the source for the edge trigger. For instruments that have an Auxiliary Input (such as the MSO58LP), AUXiliary can be selected as trigger source.

Group
Trigger

Syntax
TRIGger:{A|B}:EDGE:SOUrce {CH<x>|CH<x>_D<y>|LINE|AUXiliary}
TRIGger:{A|B}:EDGE:SOUrce?

Related Commands
TRIGger:(A|B).EDGE:SLOpe
TRIGger:(A|B).EDGE:COUPling
TRIGger:B:STATE

Arguments
CH<x> specifies an analog channel as the edge trigger source.
CH<x>_D<y> specifies a digital channel as the edge trigger source.
LINE specifies AC line voltage, and is a valid source when B trigger is inactive.
AUXiliary specifies the Auxiliary Input.

Examples

TRIGGER:A:EDGE:SOURCE CH2 sets the A edge trigger source to input channel 2.


TRIGger:{A|B}:LEVel:CH<x>

This command sets or queries the CH<x> trigger level for an Edge, Pulse Width, Runt or Rise/Fall (Transition and Slew Rate) trigger when triggering on an analog channel waveform. Each channel can have an independent trigger level. The <x> is the channel number.
Commands listed in alphabetical order

Group  Trigger

Syntax  TRIGger:{A|B}:LEVEL:CH<x> <NR3>
        TRIGger:{A|B}:LEVEL:CH<x>?

Arguments  <NR3> specifies the trigger level in user units (usually volts).

Examples  TRIGGER:A:LEVEL:CH1 1.5 sets the A trigger level for Channel 1 to 1.5 V.
          1.3000E+00, indicating that the A trigger level for Channel 2 is set to 1.3 V.

TRIGger:{A|B}:LOGIc:DELTatime

This command specifies or queries the Logic trigger delta time value. The time value is used as part of the Logic trigger condition to determine if the duration of a logic pattern meets the specified time constraints.

Group  Trigger

Syntax  TRIGger:{A|B}:LOGIC:DELTatime <NR3>
        TRIGger:{A|B}:LOGIC:DELTatime?

Arguments  <NR3> the Logic trigger delta time value.

Examples  TRIGGER:A:LOGIC:DELTatime 4.5e-9 sets the delta time to 4.5 ns.
          TRIGGER:A:LOGIC:DELTatime? might return
          :TRIGGER:A:LOGIC:DELTATIME 4.0E-9 indicating the delta time is set to 4 ns.

TRIGger:{A|B}:LOGIc:FUNCTION

This command sets or queries the logical combination of the input channels for logic triggers. This command is equivalent to selecting Logic for the Trigger Type, and setting or viewing the Define Logic.

Group  Trigger
Syntax  TRIGger:{A|B}:LOGIc:FUNCtion  {AND|NAND|NOR|OR}
        TRIGger:{A|B}:LOGIc:FUNCtion?

Arguments
AND specifies to trigger if all conditions are true.
NAND specifies to trigger if any of the conditions are false.
NOR specifies to trigger if all conditions are false.
OR specifies to trigger if any of the conditions are true.

Examples
TRIGGER:A:LOGIC:FUNCTION  AND  sets the logical combination of channels to be true when all conditions are true.
TRIGGER:A:LOGIC:FUNCTION? might return TRIGGER:A:LOGIC:FUNCTION NAND, indicating that the instrument will trigger if the AND logic conditions are false.

TRIGger:{A|B}:LOGIc:INPut:CLOCk:SOUrce

This command specifies the channel to use as the clock source for logic trigger.

Group  Trigger

Syntax  TRIGger:{A|B}:LOGIC:INPut:CLOCK:SOURce  {CH<x>|CH<x>_D<y>}
        TRIGger:{A|B}:LOGIC:INPut:CLOCK:SOURce?

Arguments
CH<x> specifies an analog channel as the clock source. Number of channels depends on instrument configuration.
CH<x>_D<y> specifies a digital channel as the clock source. Number of channels depends on instrument configuration.

Examples
TRIGger:A:LOGIC:INPut:CLOCK:SOURce  CH3 sets the clock source to channel 3.
TRIGger:{A|B}:LOGIc:POLarity

This command sets or queries the polarity for the clock channel when Use Clock Edge is set to Yes for Logic trigger type.

Group Trigger

Syntax TRIGger:{A|B}:LOGIC:POLarity {POSitive|NEGative|EITher}
TRIGger:{A|B}:LOGIC:POLarity?

Arguments NEGative specifies negative polarity.
POSITIVE specifies positive polarity.
EITHER specifies either polarity.

Examples TRIGger:A:LOGIC:POLarity EITHER sets the polarity to either positive or negative.

TRIGger:{A|B}:LOGIc:USECLockedge

This command specifies whether or not Logic trigger type uses clock source.

Group Trigger

Syntax TRIGger:{A|B}:LOGIC:USECLockedge {OFF|ON|0|1}
TRIGger:{A|B}:LOGIC:USECLockedge?

Arguments ON specifies that logic trigger type uses clock source.
OFF specifies that logic trigger type does not use clock source.
<NRL> = 0 specifies that logic trigger type does not use clock source; any other value uses clock source.

Examples TRIGger:A:LOGIC:USECLockedge OFF specifies that the clock edge will not be used.
TRIGGER:A:LOGIC:USECLOCKEDGE? might return 
:TRIGGER:A:LOGIC:USECLOCKEDGE 1 indicating that the clock edge will be used.

TRIGGER:{A|B}:LOGIC:WHEN

This command sets or queries the condition for generating an A or B logic trigger with respect to the defined input pattern. This command is equivalent to selecting Logic for Trigger Type, Use Clock Edge to No, and choosing a trigger condition from the Logic Pattern drop-down list.

Group Trigger

Syntax TRIGGER:{A|B}:LOGIC:WHEN 
{TRUE|FALSE|MOREThan|LESSThan|EQUAL|UNEQUAL} 
TRIGGER:{A|B}:LOGIC:WHEN?

Arguments

TRUE triggers on an input pattern that is true.
FALSE triggers on an input pattern that is false.
MORE than triggers on an input pattern that is true for a time period greater than a user defined Time Limit (DELTatime) value.
LESS than triggers on an input pattern that is true for a time period less than a user defined Time Limit (DELTatime) value.
EQUAL triggers on an input pattern that is true for a time period equal to a user defined Time Limit (DELTatime) value.
UNEQUAL triggers on an input pattern that is true for a time period not equal to a user defined Time Limit (DELTatime) value.

Examples TRIGGER:A:LOGIC:WHEN EQUAL specifies triggering when the input pattern is true for a time period equal to a user defined Time Limit value.

TRIGGER:{A|B}:LOGICPATTERN:{CH<x>|CH<x>_D<x>}

This command sets or queries the Logic Pattern that is used along with the Define Logic choice (LOGIC FUNCTION) to determine when the logic trigger occurs.

Group Trigger
Commands listed in alphabetical order

**Syntax**

```
TRIGger:{A|B}:LOGICPattern:{CH<x>|CH<x>_D<x>} {HIGH|LOW|X}
TRIGger:{A|B}:LOGICPattern:{CH<x>|CH<x>_D<x>}? 
```

**Arguments**

- **HIGH** specifies the logic high.
- **LOW** specifies the logic low.
- **X** specifies a don't care state.

**Examples**

```
TRIGger:A:LOGICPattern:CH1 HIGH specifies triggering on a logic high.
TRIGger:A:LOGICPattern:CH1? might return
:TRIGGER:A:LOGICPATTERN:CH1 X indicating a don't care state for channel 1.
TRIGger:A:LOGICPattern:CH1_D0 HIGH specifies a logic high.
TRIGger:A:LOGICPattern:CH1_D0? might return
:TRIGGER:A:LOGICPATTERN:CH1_D0 X indicating a don't care.
```

---

**TRIGger:{A|B}:LOWerthreshold:CH<x>**

This command sets or queries the A or B lower trigger level threshold for the channel, specified by x.

**Group**

Trigger

**Syntax**

```
TRIGger:{A|B}:LOWerthreshold:CH<x> <NR3>
TRIGger:{A|B}:LOWerthreshold:CH<x>? 
```

**Related Commands**

TRIGger:{A|B}:UPPerthreshold:CH<x>

**Arguments**

- **<NR3>** specifies the threshold voltage in user units.

**Examples**

```
TRIGGER:A:LOWERTHRESHOLD:CH2 1.3 sets the A trigger threshold voltage
for Channel 2 to 1.3 V.
TRIGGER:A:LOWERTHRESHOLD:CH3? might return
TRIGGER:A:LOWERTHRESHOLD:CH3 1.2000E+00, indicating that the A trigger
threshold voltage for Channel 3 is 1.2 V.
```
TRIGger:{A|B}:PULSEWidth:HIGHLimit

This command specifies the upper limit to use, in seconds, when triggering on detection of a pulse whose duration is inside or outside a range of two values. (Use TRIGger:{A|B}:PULSEWidth:LOWLimit to specify the lower value of the range.)

Group Trigger

Syntax

TRIGger:{A|B}:PULSEWidth:HIGHLimit <NR3>
TRIGger:{A|B}:PULSEWidth:HIGHLimit?

Related Commands

TRIGger:{A|B}:PULSEWidth:WHEN
TRIGger:{A|B}:PULSEWidth:LOWLimit

Arguments

<NR3> is a floating point number that represents the higher value of the range.

Examples

TRIGger:A:PULSEWidth:HIGHLimit 110.0E-9 sets the high limit to 110.0 ns.
TRIGger:A:PULSEWidth:HIGHLimit? might return
TRIGGER:A:PULSEWIDTH:HIGHLIMIT 178.88000E-9 indicates the high limit is set to 178.88 ns.

TRIGger:{A|B}:PULSEWidth:LOGICQUALification

This command specifies whether or not to use logic qualification for a pulse width trigger.

Group Trigger

Syntax

TRIGger:{A|B}:PULSEwidth:LOGICQUALification {ON|OFF}
TRIGger:{A|B}:PULSEwidth:LOGICQUALification?

Arguments

ON specifies that the pulse width trigger type uses logic qualification.
OFF specifies that the pulse width trigger type does not use logic qualification.

Examples

TRIGger:A:PULSEwidth:LOGICQUALification ON turns on logic qualification.
Commands listed in alphabetical order


TRIGger:{A|B}:PULSEwidth:LOWLimit

This command specifies the lower limit to use, in seconds, when triggering on detection of a pulse whose duration is inside or outside a range of two values. (Use TRIGger:{A|B}:PULSEwidth:HIGHLimit to specify the upper limit of the range.)

This command also specifies the single limit to use, in seconds, when triggering on detection of a pulse whose duration is less than, greater than, equal to, or not equal to this time limit.

Group Trigger

Syntax TRIGger:{A|B}:PULSEwidth:LOWLimit <NR3>
TRIGger:{A|B}:PULSEwidth:LOWLimit?

Related Commands TRIGger:{A|B}:PULSEwidth:WHEn
TRIGger:{A|B}:PULSEwidth:HIGHLimit

Arguments <NR3> is a floating point number that represents the lower value of the range.

Examples TRIGger:A:PULSEwidth:LOWLimit 100.0E-9 sets the low limit to 100.0 ns.
TRIGger:A:PULSEwidth:HIGHLimit? might return TRIGGER:A:PULSEWIDTH:LOWLIMIT 77.7600E-9 indicating the low limit is set to 77.76 ns.

TRIGger:{A|B}:PULSEwidth:POLarity

This command specifies the polarity for a pulse width trigger.

Group Trigger

Syntax TRIGger:{A|B}:PULSEwidth:POLarity {NEGative|POSitive}
TRIGger:{A|B}:PULSEwidth:POLarity?
Arguments

NEGative specifies a negative pulse.
POSitive specifies a positive pulse.

Examples

TRIGger:A:PULSEwidth:POLarity NEGATIVE sets the pulse polarity to negative.

TRIGger:{A|B}:PULSEWidth:SOUrce

This command specifies the source waveform for a pulse width trigger.

Group Trigger

Syntax TRIGger:{A|B}:PULSEwidth:SOUrce {CH<x>|CH<x>_D<y>}
TRIGger:{A|B}:PULSEwidth:SOUrce?

Arguments CH<x> specifies an analog input channel as the pulse-width trigger source.
CH<x>_D,y> specifies an digital input channel as the pulse-width trigger source.

Examples TRIGGER:A:PULSEWIDTH:SOURCE CH1 sets channel 1 as the pulse width source.
TRIGGER:A:PULSEWIDTH:SOURCE? might return :TRIGGER:A:PULSEWIDTH:SOURCE CH1 indicating that channel 1 is the pulse width trigger source.

TRIGger:{A|B}:PULSEWidth:WHEn

This command specifies to trigger when a pulse is detected with a width (duration) that is less than, greater than, equal to, or unequal to a specified value (set using TRIGger:(A|B):PULSEWidth LOWLimit), OR whose width falls outside of or within a specified range of two values (set using TRIGger:(A|B):PULSEWidth LOWLimit and TRIGger:(A|B):PULSEWidth HIGHLimit).

Group Trigger
Syntax

TRIGger:{A|B}:PULSEwidth:WHEn
{LESSthan|MOREthan|EQUAL|UNEQual|WIThIn|OUTside}

TRIGger:{A|B}:PULSEwidth:WHEn?

Related Commands

TRIGger:(A|B):PULSEWidth LOWLimit
TRIGger:(A|B):PULSEWidth HIGHLimit
TRIGger:(A|B):PULSEWidth SOURce

Arguments

LESSthan causes a trigger when a pulse is detected with a width less than the time set by the TRIGger:(A|B):PULSEWidth LOWLimit command.

MOREthan causes a trigger when a pulse is detected with a width greater than the time set by the TRIGger:(A|B):PULSEWidth LOWLimit command.

EQUAL causes a trigger when a pulse is detected with a width equal to the time period specified in TRIGger:(A|B):PULSEWidth LOWLimit within a ±5% tolerance.

UNEQual causes a trigger when a pulse is detected with a width greater than or less than (but not equal) the time period specified in TRIGger:(A|B):PULSEWidth LOWLimit within a ±5% tolerance.

WIThIn causes a trigger when a pulse is detected that is within a range set by two values.

OUTside causes a trigger when a pulse is detected that is outside of a range set by two values.

Examples

:TRIGger:B:PULSEwidth:WHEn LESSthan causes the oscilloscope to trigger when a pulse is detected that is shorter than the time specified by TRIGger:(A|B):PULSEWidth LOWLimit.

TRIGger:A:PULSEwidth:WHEn ? might return TRIGGER:A:PULSEWIDTH:WHEN GREATER THAN 2.0000E-9, indicating that a trigger is generated when a pulse is detected greater than 2 ns.

TRIGger:B:PULSEwidth:WHEn? might return TRIGger:B:PULSEwidth:WHEn MOREthan indicating that a trigger is generated when a pulse is detected that is greater than the time specified by TRIGger:(A|B):PULSEWidth LOWLimit.

TRIGger:{A|B}:RUNT:LOGICQUALification

This command specifies whether or not to use logic qualification for a runt trigger.

Group Trigger
Commands listed in alphabetical order

**TRIGger:{A|B}:RUNT:LOGICQUALification**

Syntax

```
TRIGger:{A|B}:RUNT:LOGICQUALification {ON|OFF}
TRIGger:{A|B}:RUNT:LOGICQUALification?
```

Arguments

- **ON** specifies that the runt trigger type uses logic qualification.
- **OFF** specifies that the runt trigger type does not use logic qualification.

Examples

```
TRIGger:A:RUNT:LOGICQUALification ON turns on logic qualification.
```

**TRIGger:{A|B}:RUNT:POLarity**

This command specifies the polarity for the runt trigger.

Syntax

```
TRIGger:{A|B}:RUNT:POLarity {EITher|NEGative|POSitive}
TRIGger:{A|B}:RUNT:POLarity?
```

Arguments

- **POSitive** indicates that the rising edge crosses the low threshold and the falling edge recrosses the low threshold without either edge ever crossing the high threshold.
- **NEGative** indicates that the falling edge crosses the high threshold and the rising edge recrosses the high threshold without either edge ever crossing the low threshold.
- **EITher** triggers on a runt of either polarity.

Examples

```
TRIGger:A:RUNT:POLarity NEGATIVE specifies that the polarity of the A pulse runt trigger is negative.
```

**TRIGger:{A|B}:RUNT:SOUrce**

This command specifies the source waveform for the runt trigger.

**NOTE.** Digital channels are not supported as runt trigger sources.
Commands listed in alphabetical order

Group  Trigger

Syntax  TRIGger:{A|B}:RUNT:SOURcE {CH<x>}
       TRIGger:{A|B}:RUNT:SOURcE?

Arguments  CH<x> specifies the analog channel number to use as the source
waveform for the runt trigger. To specify the threshold levels when using
CH<x> as the source, use TRIGger:(A|B):LOWerthreshold CH<x> and
TRIGger:(A|B):UPPerthreshold CH<x>.

Examples  TRIGger:A:RUNT:SOURcE CH4 sets channel 4 as the trigger source.
indicating that channel 2 is the trigger source.

TRIGger:{A|B}:RUNT:WHEn

This command specifies the type of pulse width the trigger checks for when
it detects a runt.

Group  Trigger

Syntax  TRIGger:{A|B}:RUNT:WHEn
       {LESSthan|MOREthan|EQUAL|UNEQual|OCCURS}
       TRIGger:{A|B}:RUNT:WHEn?

Related Commands  TRIGger:(A|B):RUNT:WIDth

Arguments  OCCURS argument specifies a trigger event if a runt of any detectable width occurs.
LESSthan argument sets the oscilloscope to trigger if the a runt pulse is detected
with width less than the time set by the TRIGger:(A|B):RUNT:WIDth command.
MOREthan argument sets the oscilloscope to trigger if the a runt pulse is detected
with width greater than the time set by the TRIGger:(A|B):RUNT:WIDth command.
EQUAL argument sets the oscilloscope to trigger if a runt pulse is detected with
width equal to the time period specified in TRIGger:(A|B):RUNT:WIDth within
a ±5% tolerance.
NOTEQUAL argument sets the oscilloscope to trigger if a runt pulse is detected with
width greater than or less than (but not equal to) the time period specified in
TRIGger:(A|B):RUNT:WIDth within a ±5% tolerance.
Examples

TRIGger:A:RUNT:WHen MORETHAN sets the runt trigger to occur when the oscilloscope detects a runt in a pulse wider than the specified width.


**TRIGger:{A|B}:RUNT:WIDth**

This command specifies the width, in seconds, for a runt trigger.

**Group** Trigger

**Syntax** TRIGger:{A|B}:RUNT:WIDth <NR3>

TRIGger:{A|B}:RUNT:WIDth?

**Related Commands** TRIGger(A|B):RUNT:WHen

**Arguments**

<NR3> is a floating point number that specifies the minimum width, in seconds.

**Examples**

TRIGger:A:RUNT:WIDth 15E-6 sets the minimum width of the pulse runt trigger to 15 µs.


**TRIGger:{A|B}:SETHold:CLOCk:EDGE**

This command specifies the clock edge polarity for setup and hold triggering.

**Group** Trigger

**Syntax** TRIGger:{A|B}:SETHold:CLOCk:EDGE {FALL|RISe}

TRIGger:{A|B}:SETHold:CLOCk:EDGE?

**Arguments**

FALL specifies polarity as the clock falling edge.

RISe specifies polarity as the clock rising edge.
Examples

TRIGger:A:SETHold:CLOCK:EDGE RISE specifies the polarity as the clock rising edge.

TRIGger:A:SETHold:CLOCK:EDGE? might return
:TRIGGER:A:SETHOLD:CLOCK:EDGE RISE indicating that polarity is specified as the clock rising edge.

TRIGger:{A|B}:SETHold:CLOCK:SOUrce

This command specifies the clock source for the setup and hold triggering. You cannot specify the same source for both clock and data.

Group

Trigger

Syntax

TRIGger:{A|B}:SETHold:CLOCK:SOURce {CH<x>|CH<x>_D<y>}

TRIGger:{A|B}:SETHold:CLOCK:SOURce?

Arguments

CH<x> specifies the analog channel to use as the clock source waveform.
CH<x>_D<y> specifies the digital channel to use as the clock source waveform.

Examples

TRIGger:A:SETHold:CLOCK:SOURce CH1 specifies channel 1 as the clock source for a setup and hold trigger operation.

TRIGger:A:SETHold:CLOCK:SOURce? might return
:TRIGGER:A:SETHOLD:CLOCK:SOURce CH4 indicating that channel 4 is the clock source for a setup and hold trigger operation.

TRIGger:{A|B}:SETHold:HOLDTime

This command specifies the hold time for setup and hold violation triggering. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then setting the desired Hold Time.

Group

Trigger

Syntax

TRIGger:{A|B}:SETHold:HOLDTime <NR3>

TRIGger:{A|B}:SETHold:HOLDTime?
Arguments  

<NR3> is a floating point number that specifies the hold time setting, in seconds. Positive values for hold time occur after the clock edge. Negative values occur before the clock edge.

Examples  

TRIGger:A:SETHold:HOLDTime 3.0E-3 sets the hold time for the setup and hold trigger to 3 ms.

TRIGger:A:SETHold:HOLDTime? might return :TRIGGER:A:SETHOLD:HOLDTime 2.0000E-09 indicating that the current hold time for the setup and hold trigger is 2 ns.

TRIGger:{A|B}:SETHold:SETTime  

This command specifies the setup time for setup and hold violation triggering. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then setting the desired Setup Time.

Group  

Trigger

Syntax  

TRIGger:{A|B}:SETHold:SETTime <NR3>
TRIGger:{A|B}:SETHold:SETTime?

Arguments  

<NR3> is a floating point number that specifies the setup time for setup and hold violation triggering.

Examples  

TRIGger:A:SETHold:SETTime 3.0E-6 specifies that the current setup time for setup and hold trigger is 3 µs.


TRIGger:{A|B}:SETHOLDLogicval:{CH<x>|CH<x>_D<x>}

This command sets or queries whether the specified channel is included (INCLude) or not included (DONTInclude) in the Setup & Hold trigger input configuration. The channel number is specified by <x>.

Group  

Trigger
**TRIGGER:{A|B}:SETHOLDLogicval:{CH<x>|CH<x>_D<x>}**

**Syntax**

`TRIGGER:{A|B}:SETHOLDLogicval:{CH<x>|CH<x>_D<x>}`

**Arguments**

- **INCLUDE** specifies including the specified channel in the Setup & Hold trigger input configuration.
- **DONTInclude** specifies not including the specified channel in the Setup & Hold trigger input configuration.

**Examples**

- `TRIGGER:A:SETHOLDLogicval:CH1 INCLUDE` specifies including the specified channel in the Setup & Hold trigger input configuration.
- `TRIGGER:A:SETHOLDLogicval:CH1?` might include `:TRIGGER:A:SETHOLDLOGICVAL:CH1 DONTINCLUDE` indicating not to include the channel in the configuration.
- `TRIGGER:A:SETHOLDLogicval:CH1_D0` include specifies including the specified channel in the setup and hold trigger input configuration.
- `TRIGGER:A:SETHOLDLogicval:CH1_D0?` might return `:TRIGGER:A:SETHOLDLOGICVAL:CH1_D0 DONTINCLUDE` indicating the channel will not be included in the configuration.

**TRIGGER:{A|B}:TIMEOut:LOGICQUALification**

This command specifies whether or not to use logic qualification for a timeout trigger.

**Group**

Trigger

**Syntax**

`TRIGGER:{A|B}:TIMEOut:LOGICQUALification {ON|OFF}`

**Arguments**

- **ON** specifies that the timeout trigger type uses logic qualification.
- **OFF** specifies that the timeout trigger type does not use logic qualification.

**Examples**

- `TRIGGER:A:TIMEOut:LOGICQUALification ON` specifies using logic qualification.
**TRIGger:{A|B}:TIMEOut:POLarity**

When triggering using the TIMEOut trigger type, this command specifies the polarity to be used.

**Group** Trigger

**Syntax**

```
TRIGger:{A|B}:TIMEOut:POLarity {STAYSHigh|STAYSLow|EITher}
TRIGger:{A|B}:TIMEOut:POLarity?
```

**Related Commands**

- TRIGger:{A|B}:TIMEOut:SOURce
- TRIGger:{A|B}:TIMEOut:TIMe

**Arguments**

- **STAYSHigh** — Trigger when the signal stays high during the timeout time specified by the command TRIGger:{A|B}:TIMEOut:TIMe.
- **STAYSLow** — Trigger when the signal stays low during the timeout time specified by the command TRIGger:{A|B}:TIMEOut:TIMe.
- **EITher** — Trigger when the signal is either high or low during the timeout time specified by the command TRIGger:{A|B}:TIMEOut:TIMe.

**Examples**

```
TRIGger:A:TIMEOut:POLarity STAYSHigh specifies to trigger when the signal stays high during the timeout time when triggering using the TIMEOut trigger type.
TRIGger:A:TIMEOut:POLarity? might return EITHER, indicating that the signal stays either high or low during the timeout time.
```

---

**TRIGger:{A|B}:TIMEOut:SOURce**

When triggering using the TIMEOut trigger type, this command specifies the source. The available sources are live channels and digital channels. The default is channel 1. The timeout trigger type is selected using TRIGger:{A|B}:TYPE:TIMEOut.

**Group** Trigger

**Syntax**

```
TRIGger:{A|B}:TIMEOut:SOURce {CH<x>|CH<x>_D<y>}
TRIGger:{A|B}:TIMEOut:SOURce?
```

**Related Commands**

- SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOURce
TRIGger\{(A|B)\}:TIMEOut:TIMe
TRIGger\{(A|B)\}:TIMEOut:POLarity

Arguments
CH<x> specifies an analog channel as the timeout trigger source.
CH<x>_D<y> specifies a digital channel as the timeout trigger source.

Examples
TRIGger:A:TIMEOut:SOURce CH4 specifies to use channel 4 as the source for the timeout trigger.
TRIGger:A:TIMEOut:SOURce? might return CH1, indicating that channel 1 has been set as the source for the timeout trigger.

TRIGger\{(A|B)\}:TIMEOut:TIMe

When triggering using the TIMEOut trigger type, this command specifies the timeout time, in seconds. This command is equivalent to selecting Timeout from the Trig menu and setting a value for Time Limit. The timeout trigger type is selected using TRIGger\{(A|B)\}:TYPe TIMEOut

Group Trigger

Syntax TRIGger\{(A|B)\}:TIMEOut:TIMe <NR3>
TRIGger\{(A|B)\}:TIMEOut:TIMe?

Related Commands TRIGger\{(A|B)\}:TIMEOut:POLarity
TRIGger\{(A|B)\}:TIMEOut:SOURce

Arguments <NR3> is a floating point number that specifies the timeout time, in seconds.

Examples :TRIGger:A:TIMEOut:TIME 4.0E-9 specifies the timeout time of 4.0 nsec.

TRIGger\{(A|B)\}:TRANsition:DELTatime

This command specifies the delta time (that is Time Limit) used in calculating the transition value for the transition (that is Rise or Fall Time) trigger.

Group Trigger
Syntax  TRIGger:{A|B}:TRANSition:DELTatime  <NR3>
TRIGger:{A|B}:TRANSition:DELTatime?

Arguments  <NR3> is a floating point number that specifies the delta time, in seconds.

Examples  TRIGGER:A:TRANSITION:DELTATIME  15E-6 sets the delta time of the transition trigger to 15 µs.

TRIGger:{A|B}:TRANSition:LOGICQUALification

This command specifies whether or not to use logic qualification for a transition trigger.

Group      Trigger

Syntax  TRIGger:{A|B}:TRANSition:LOGICQUALification {ON|OFF}
TRIGger:{A|B}:TRANSition:LOGICQUALification?

Arguments  ON specifies that the transition trigger type uses logic qualification.
OFF specifies that the transition trigger type does not use logic qualification.

Examples  TRIGger:A:TRANSITION:LOGICQUALification ON specifies using logic qualification.

TRIGger:{A|B}:TRANSition:POLarity

This command specifies the polarity for the transition trigger.

Group      Trigger

Syntax  TRIGger:{A|B}:TRANSition:POLarity {EITHER|NEGative|POSitive}
TRIGger:{A|B}:TRANSition:POLarity?

Arguments

POSitive indicates that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for transition triggering to occur.

NEGative indicates that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.

EITHER indicates either positive or negative polarity.

Examples

TRIGGER:A:TRANSITION:POLARITY NEGATIVE sets the transition polarity to negative.

TRIGGER:A:TRANSITION:POLARITY? might return :TRIGGER:A:TRANSITION:POLARITY EITHER indicating that the polarity can be either positive or negative.

TRIGger:{A|B}:TRANSition:SOUrce

This command specifies the source waveform for a transition trigger.

NOTE. Digital channels are not supported as transition trigger sources.

Group Trigger

Syntax TRIGger:{A|B}:TRANSition:SOUrce {CH<x>}
TRIGger:{A|B}:TRANSition:SOUrce?

Arguments CH<x> specifies one of the analog channels to be used as the source for a transition trigger.

Examples TRIGGER:A:TRANSITION:SOURCE CH4 sets channel 4 as the source for the transition trigger.


TRIGger:{A|B}:TRANSition:WHEn

This command specifies whether to check for a transitioning signal that is faster or slower than the specified delta time.
Commands listed in alphabetical order

**Group**   
Trigger

**Syntax**

TRIGger:{A|B}:TRANSition:WHEn {SLOWER|FASTER|EQUAL|UNEQUAL}

TRIGger:{A|B}:TRANSition:WHEN?

**Arguments**

FASTER sets the trigger to occur when the signal transition time is faster than the time set by TRIGger:A:TRANSition:DELTatime.

SLOWER sets the trigger to occur when the signal transition time is slower than the time set by TRIGger:A:TRANSition:DELTatime.

EQUAL sets the trigger to occur when the signal transition time is equal to the time set by TRIGger:A:TRANSition:DELTatime.

UNEQUAL sets the trigger to occur when the signal transition time is not equal to the time set by TRIGger:A:TRANSition:DELTatime.

**Examples**

TRIGGER:A:TRANSITION:WHEN SLOWER sets the trigger to occur when the signal transition time is slower than the time set by the TRIGger:A:TRANSition:DELTatime command.


**TRIGger:{A|B}:TYPE**

This command sets or queries the type of A or B trigger.

**Group**   
Trigger

**Syntax**

TRIGger:{A|B}:TYPE {EDGE|WIDth|TIMEOut|RUNt|WINdow|LOGIC|SETHold|TRANSition|BUS}

TRIGger:{A|B}:TYPE?

**Arguments**

EDGE is a normal trigger. A trigger event occurs when a signal passes through a specified voltage level in a specified direction and is controlled by the TRIGger:A:EDGE commands.

WIDth specifies that the trigger occurs when a pulse with a specified width is found.

TIMEOut specifies that a trigger occurs when a pulse with the specified timeout is found.
Commands listed in alphabetical order

RUNT specifies that a trigger occurs when a pulse with the specified parameters is found.

WINDOW specifies that a trigger occurs when a signal with the specified window parameters is found.

LOGIC specifies that a trigger occurs when specified conditions are met and is controlled by the TRIGGER:{A|B}:LOGIC commands.

SETHold specifies that a trigger occurs when a signal is found that meets the setup and hold parameters.

Transition specifies that a trigger occurs when a specified pulse is found that meets the transition trigger parameters.

BUS specifies that a trigger occurs when a signal is found that meets the specified bus setup parameters.

Examples

TRIGGER:A:TYPE EDGE sets the A trigger type to EDGE.

TRIGGER:A:TYPE? might return :TRIGGER:A:TYPE RUNT indicating that the A trigger type is a runt trigger.

TRIGGER:{A|B}:UPPerthreshold:CH<x>

This command sets or queries the specified channel upper trigger level. The CH<x> range is 1 to 8 and depends on the number of analog channels on your instrument.

Group Trigger

Syntax TRIGGER:{A|B}:UPPerthreshold:CH<x> <NR3>
TRIGGER:{A|B}:UPPerthreshold:CH<x>? Related Commands TRIGGER:{A|B}:LOWerthreshold:CH<x>

Arguments <NR3> specifies the trigger level in user units (usually volts).

Examples TRIGGER:A:UPPERTHRESHOLD:CH1 1.3 This command sets the A trigger level for Channel 1 to 1.3 V level.

TRIGGER:A:UPPERTHRESHOLD:CH2? might return TRIGGER:A:UPPERTHRESHOLD:CH2 1.3000E+00, indicating that the A trigger level for Channel 2 is set to 1.3 V.
**TRIGger:{A|B}:WINdow:CROSSIng**

This command sets or queries the window trigger threshold crossing of the selected trigger source. The threshold crossing selection is only effective when `:TRIGger:{A|B}:WINdow:WHen` is `INSIDEGreater` or `OUTSIDEGreater`.

<table>
<thead>
<tr>
<th>Group</th>
<th>Trigger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>`TRIGger:{A</td>
</tr>
</tbody>
</table>

**Arguments**

- **UPPer** if `:TRIGger:{A|B}:WINdow:WHen` is `INSIDEGreater`, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) and then exits through the upper threshold. If `:TRIGger:{A|B}:WINdow:WHen` is `OUTSIDEGreater`, the instrument triggers when the signal remains above the upper threshold for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) and then crosses downward through the upper threshold.

- **LOWer** if `:TRIGger:{A|B}:WINdow:WHen` is `INSIDEGreater`, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) and then exits through the lower threshold. If `:TRIGger:{A|B}:WINdow:WHen` is `OUTSIDEGreater`, the instrument triggers when the signal remains below the lower threshold for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) and then crosses upwards through the lower threshold.

- **EITher** if `:TRIGger:{A|B}:WINdow:WHen` is `INSIDEGreater`, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) and then exits through either the upper or lower threshold. If `:TRIGger:{A|B}:WINdow:WHen` is `OUTSIDEGreater`, the instrument triggers when the signal remains either above the upper threshold or below the lower threshold for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) and then crosses a threshold.

- **NONE** if `:TRIGger:{A|B}:WINdow:WHen` is `INSIDEGreater`, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) without crossing through either the upper or lower threshold. If `:TRIGger:{A|B}:WINdow:WHen` is `OUTSIDEGreater`, the instrument triggers when the signal remains outside the upper and lower thresholds for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) without crossing through either the upper or lower threshold.
**Examples**

TRIGger:A:WINdow:CROSSIng LOWER sets crossing to lower.

**TRIGger:{A|B}:WINdow:LOGICQUALification**

This command specifies whether or not to use logic qualification for a window trigger.

**Group** Trigger

**Syntax**

TRIGger:{A|B}:WINdow:LOGICQUALification {ON|OFF}
TRIGger:{A|B}:WINdow:LOGICQUALification?

**Arguments**

ON specifies that the window trigger type uses logic qualification.
OFF specifies that the window trigger type does not use logic qualification.

**Examples**

TRIGger:A:WINdow:LOGICQUALification ON turns on logic qualification.

**TRIGger:{A|B}:WINdow:SOUrce**

This command sets or queries the source for a window trigger.

*NOTE. Digital channels are not supported as window trigger sources.*

**Group** Trigger

**Syntax**

TRIGger:{A|B}:WINdow:SOUrce {CH<x>}
TRIGger:{A|B}:WINdow:SOUrce?

**Arguments**

The window trigger source channel.
Examples

TRIGger:A:WINdow:SOUrce CH2 sets the source to channel 2.


**TRIGger:{A|B}:WINdow:WHEn**

This command sets or queries the window trigger event. This command is equivalent to selecting Window Setup from the Trig menu and selecting from the Window Trigger When box.

**Group** Trigger

**Syntax**

TRIGger:{A|B}:WINdow:WHEn {ENTERSWindow|EXITSWindow|INSIDEGreater|OUTSIDEGreater}

TRIGger:{A|B}:WINdow:WHEn?

**Arguments**

OUTSIDEGreater specifies a trigger event when the signal leaves the window defined by the threshold levels for the time specified by Width.

INSIDEGreater specifies a trigger event when the signal enters the window defined by the threshold levels for the time specified by Width.

ENTERSWindow specifies a trigger event when the signal enters the window defined by the threshold levels.

EXITSWindow specifies a trigger event when the signal leaves the window defined by the threshold levels.

**Examples**

TRIGger:A:WINdow:WHEn EXITSWindow specifies a trigger event when the signal leaves the window defined by the threshold levels.


**TRIGger:{A|B}:WINdow:WIDth**

This command sets or queries the minimum width for a window violation. This command is equivalent to selecting Window Setup from the Trig menu, selecting Inside > Limit or Outside > Limit in the Trigger When box, and setting the Time Limit.

**Group** Trigger
Commands listed in alphabetical order

**Syntax**

TRIGger:{A|B}:WINdow:WIDth <NR3>

**Arguments**

<NR3> is the minimum width for a window violation.

**Examples**

TRIGger:A:WINDOW:WIDTH 4.5e-9 sets the minimum width for a window violation to 4.5 ns.

TRIGger:A:WINDOW:WIDTH? might return :TRIGGER:A:WINDOW:WIDTH 4.0E-9 indicating that 4.0 ns is the minimum width for a window violation.

**TRIGger:A:HOLDoFF:BY**

This command sets or queries the type of holdoff for the A trigger. Holdoff types are expressed as either user-specified time (TIMe) or by an internally calculated random time value (RANDom). This command is equivalent to selecting Mode & Holdoff from the Trig menu and then setting the Holdoff type.

**Group**

Trigger

**Syntax**

TRIGger:A:HOLDoff:BY {TIMe|RANDom}

TRIGger:A:HOLDoff:BY?

**Related Commands**

TRIGger:A:HOLDoff:TIMe

**Arguments**

TIMe enables you to set the holdoff time via the TRIGger:A:HOLDoff:TIMe command.

RANDom specifies a random time value.

**Examples**

TRIGGER:A:HOLDOFF:BY TIME sets the holdoff to the "by time" setting. This enables you to set the holdoff time.


**TRIGger:A:HOLDoFF:TIMe**

This command sets or queries the A trigger holdoff time. This command is equivalent to selecting Mode & Holdoff from the Trig menu, selecting Time, and then setting the desired Holdoff Time.
### TRIGGER:A:HOLDoff:TIME

**Syntax**

- TRIGger:A:HOLDoff:TIME <NR3>
- TRIGger:A:HOLDoff:TIME?

**Arguments**

<NR3> specifies the holdoff time in seconds. The range is from 0 seconds through 10 seconds.

**Examples**

- TRIGGER:A:HOLDoff:TIME 10 sets the A trigger holdoff time to 10 s.

### TRIGger:A:LOGICQUALification

This command sets or queries the type of logic qualification to perform.

**Syntax**

- TRIGger:A:LOGICQUALification {AND|OR|NAND|NOR}
- TRIGger:A:LOGICQUALification?

**Arguments**

- AND specifies to trigger if all conditions are true.
- NAND specifies to trigger if any of the conditions are false.
- NOR specifies to trigger if all conditions are false.
- OR specifies to trigger if any of the conditions are true.

**Examples**

- TRIGGER:A:LOGICQUALIFICATION OR specifies to trigger if any of the conditions are true.
- TRIGGER:A:LOGICQUALIFICATION? might return TRIGGER:A:LOGICQUALIFICATION AND indicating the instrument will trigger if all conditions are true.
TRIGger:A:MODE

This command sets or queries the A trigger mode. This command is equivalent to pushing the Mode button on the front panel.

Group  Trigger

Syntax  TRIGger:A:MODE {AUTO|NORMal}
        TRIGger:A:MODE?

Arguments  AUTO generates a trigger if one is not detected within a specified time period.
           NORMal waits for a valid trigger event.

Examples  TRIGGER:A:MODE NORMAL sets the trigger mode to normal.
          TRIGGER:A:MODE? might return TRIGGER:A:MODE AUTO, indicating that the trigger mode is auto.

TRIGger:AUXLevel

For those instruments that have an Auxiliary Input (such as an MSO58LP), this command sets or queries the Auxiliary Input voltage level to use for an edge trigger.

Group  Trigger

Syntax  TRIGger:AUXLevel {<NR3>|ECL|TTL}
        TRIGger:AUXLevel?

Arguments  <NR3> is trigger level in Volts.
           ECL sets trigger level to -1.3 Volts.
           TTL sets trigger level to 1.4 Volts.

Returns  <NR3> is trigger level in Volts.

Examples  TRIGger:AUXLevel 1.2 sets trigger level for the Auxiliary Input to 1.2 Volts.
**TRIGger:B:BY**

This command selects or returns whether the B trigger occurs after a specified number of events or a specified period of time after the A trigger.

**Group**  
Trigger

**Syntax**  
TRIGger:B:BY {EVENTS|TIME}  
TRIGger:B:BY?

**Related Commands**  
TRIGger:B:EVENTS:COUNt  
TRIGger:B:TIME  
TRIGger:B:STATE

**Arguments**  
EVENTS sets the B trigger to take place following a set number of trigger events after the A trigger occurs. The number of events is specified by TRIGger:B:EVENTS:COUNt.

TIME sets the B trigger to occur a set time after the A trigger event. The time period is specified by TRIGger:B:TIME.

**Examples**  
TRIGGER:B:BY TIME sets the B trigger to occur at a set time after the A trigger event.

TRIGGER:B:BY? might return TRIGGER:B:BY EVENTS, indicating that the B trigger takes place following a set number of trigger events after the A trigger occurs.

**TRIGger:B:EVENTS:COUNt**

This command sets or queries the number of events that must occur before the B trigger. The B trigger event count applies only if TRIGger:B:BY is set to EVENTS.

**Group**  
Trigger

**Syntax**  
TRIGger:B:EVENTS:COUNt <NR1>  
TRIGger:B:EVENTS:COUNt?

**Related Commands**  
TRIGger:B:STATE
## TRIGger:B:EVENTS:COUNT

Arguments

<NR1> is the number of B trigger events, which can range from 1 to 65,471.

Examples

TRIGGER:B:EVENTS:COUNT 4 sets the number of B trigger events to four.

TRIGGER:B:EVENTS:COUNT? might return TRIGGER:B:EVENTS:COUNT 2, indicating that two events must occur after the A trigger before the B trigger can occur.

## TRIGger:B:RESET (No Query Form)

This command sets the B reset trigger level to 50%.

**Group**

Trigger

**Syntax**

TRIGger:B:RESET SETLevel

**Arguments**

SETLevel sets the B reset trigger level to 50%.

**Examples**

TRIGger:B:RESET SETLevel sets the B reset trigger level to 50%.

## TRIGger:B:RESET:EDGE:COUPling

Sets or queries the trigger coupling for a sequential edge trigger reset when the Source is set to an analog channel.

**Group**

Trigger

**Syntax**

TRIGger:B:RESET:EDGE:COUPling {DC|HFRej|LFRej|NOISErej}

TRIGger:B:RESET:EDGE:COUPling?

**Related Commands**

**Arguments**

DC selects DC trigger coupling.

HFRej selects high frequency low sensitivity.

LFRej selects low frequency low sensitivity.

NOISERej selects DC low sensitivity.
Examples

TRIGGER:B:RESET:EDGE:COUPLING HFRLFJ sets high frequency low
sensitivity.

TRIGGER:B:RESET:EDGE:COUPLING? might return
TRIGGER:B:RESET:EDGE:COUPLING DC indicating DC trigger coupling
is selected.

TRIGger:B:RESET:EDGE:LEVel

This command sets the voltage level to use for an Edge Reset trigger when
triggering on an analog channel waveform.

Group
Trigger

Syntax
TRIGger:B:RESET:EDGE:LEVel <NR3>

Arguments
<NR3> is the voltage level to use for an Edge Reset trigger when triggering on
an analog channel waveform.

Examples
TRIGGER:B:RESET:EDGE:LEVEL 50.0e-3 sets the level to 50.0 mV.
TRIGGER:B:RESET:EDGE:LEVEL? might return
TRIGGER:B:RESET:EDGE:LEVEL 0.0E+0 indicating the level is set to 0.0 V.

TRIGger:B:RESET:EDGE:SLOpe

This command sets or queries the trigger slope for a sequential edge trigger reset.

Group
Trigger

Syntax
TRIGger:B:RESET:EDGE:SLOpe {RISe|FALL|EITHER}
TRIGger:B:RESET:EDGE:SLOpe?

Arguments
RISe specifies to reset the trigger on the rising or positive edge of a signal.
FALL specifies to reset the trigger on the falling or negative edge of a signal.
EITHER specified to reset the trigger on either the rising or falling edge of a signal.

Examples
TRIGGER:B:RESET:EDGE:SLOPE FALL specifies to reset the trigger on the
falling or negative edge of a signal.
TRIGGER:B:RESET:EDGE:SLOPE? might return
TRIGGER:B:RESET:EDGE:SLOPE RISE indicating the instrument is set to reset
the trigger on the rising or positive edge of a signal.

TRIGGER:B:RESET:EDGE:SOURce

This command sets or queries the trigger source for the A→B sequential edge
trigger reset feature.

Group  Trigger

Syntax  
TRIGGER:B:RESET:EDGE:SOURce {CH<x>|CH<x>_D<y>}
TRIGGER:B:RESET:EDGE:SOURce?

Arguments  The source channel for the trigger reset.

Examples  TRIGGER:B:RESET:EDGE:SOURce CH4 sets Channel 4 as the input source for
the trigger reset.

TRIGGER:B:RESET:EDGE:SOURce? might return
TRIGGER:B:RESET:EDGE:SOURce CH1, indicating that the current input
source for the trigger reset is Channel 1.

TRIGGER:B:RESET:TIMEOut:TIMe

This command sets or queries the reset timer for a sequential timeout trigger reset.

Group  Trigger

Syntax  
TRIGGER:B:RESET:TIMEOut:TIMe <NR3>
TRIGGER:B:RESET:TIMEOut:TIMe?

Arguments  <NR3> is the reset timer for a sequential timeout trigger reset.

Examples  TRIGGER:B:RESET:TIMEOUT:TIME 100e-9 sets the time to 100 ns.
TRIGGER:B:RESET:TIMEOUT:TIME? might return
TRIGGER:B:RESET:TIMEOUT:TIME 20.0E-9 indicating the timeout time
is set to 20 ns.
TRIGGER:B:RESET:TYPE

This command sets or queries the type of A→B sequential trigger reset. If the B trigger reset is active, the reset criteria are part of the B triggering sequence. If the reset conditions defined by the reset type are met, the instrument must start over searching for a new occurrence of the A event.

You must identify a trigger Source and Threshold for each reset type, except for the Timeout trigger type.

**NOTE.** *If a reset condition occurs, the reset criteria itself is reset and must start over.*

<table>
<thead>
<tr>
<th>Group</th>
<th>Trigger</th>
</tr>
</thead>
</table>
| Syntax  | TRIGGER:B:RESET:TYPE {NONE|TIMEOut|EDGE}  
TRIGGER:B:RESET:TYPE? |
| Arguments | NONE defeats the trigger reset feature.  
TIMEOut initiates a reset if the timeout conditions specified by TRIGGER:B:RESET:TIMEOut are met.  
EDGE initiates a reset if the edge trigger conditions are met. |
| Examples | TRIGGER:B:RESET:TYPE NONE deactivates the sequential trigger reset.  
TRIGGER:B:RESET:TYPE? This query might return TRIGGER:B:RESET:TYPE TIMEOUT, indicating that the sequential trigger reset is active following a timeout. |

TRIGGER:B:STATE

This command sets or queries the state of B trigger activity. If the B trigger state is on, the B trigger is part of the triggering sequence. If the B trigger state is off, then only the A trigger causes the trigger event.

<table>
<thead>
<tr>
<th>Group</th>
<th>Trigger</th>
</tr>
</thead>
</table>
| Syntax  | TRIGGER:B:STATE {ON|OFF|<NR1>}  
TRIGGER:B:STATE? |
| Related Commands | TRIGGER:A:MODE |
Arguments
ON indicates that the B trigger is active and causes trigger events with the A trigger.
OFF indicates that only the A trigger causes trigger events.
<NR1> is an integer number. 0 turns off the B trigger; any other value activates the B trigger.

Examples
TRIGGER:B:STATE ON sets the B trigger to active, making it capable of causing trigger events.
TRIGGER:B:STATE? might return TRIGGER:B:STATE 0, indicating that the B trigger is inactive and that only the A trigger causes trigger events.

TRIGger:B:TIMe
This command sets or queries B trigger delay time, in seconds. The B Trigger time applies only if TRIGger:B:BY is set to TIMe.

Group Trigger

Syntax
TRIGGER:B:TIME <NR3>
TRIGGER:B:TIME?

Related Commands
TRIGger:B:BY
TRIGger:B:EVENTS:COUNt
TRIGger:B:STATE

Arguments
<NR3> is the B trigger delay time in seconds.

Examples
TRIGGER:B:TIME 4E-6 sets the B trigger delay time to 4 µs.
TRIGGER:B:TIME? might return TRIGGER:B:TIME 16.0000E-9, indicating that the B trigger time is set to 16 ns.

TRIGger:STATE? (Query Only)
This query-only command returns the current state of the triggering system.

Group Trigger
### TRIGger:STATE?

**Syntax**
```
TRIGger:STATE?
```

**Related Commands**
```
TRIGger:A:MODe
```

**Returns**
- **ARMED** indicates that the instrument is acquiring pretrigger information.
- **AUTO** indicates that the instrument is in the automatic mode and acquires data even in the absence of a trigger.
- **READY** indicates that all pretrigger information is acquired and that the instrument is ready to accept a trigger.
- **SAVE** indicates that the instrument is in save mode and is not acquiring data.
- **TRIGGER** indicates that the instrument triggered and is acquiring the post trigger information.

**Examples**
```
TRIGGER:STATE? might return TRIGGER:STATE ARMED, indicating that the pretrigger data is being acquired.
```

### *TST? (Query Only)

**Syntax**
```
*TST?
```

**Group**
Miscellaneous

**Examples**
```
*TST? always returns 0.
```

### UNDO (No Query Form)

**Syntax**
```
UNDO
```

**Group**
Miscellaneous

**Examples**
```
UNDO reverts the instrument settings to a state before the previous command or user interface action.
```
Examples
UNDO reverts the instrument settings to a state before the previous command or user interface action.

UNLock (No Query Form)

This command (no query form) unlocks the front panel. The command is equivalent to LOCk NONe.

**NOTE.** If the instrument is in the Remote With Lockout State (RWLS), the UNLock command has no effect. For more information, see the ANSI-IEEE Std 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.

Group      Miscellaneous
Syntax     UNLock ALL
Related Commands  LOCk
Arguments  ALL specifies that all front panel buttons and knobs are unlocked.
Examples  UNLOCK ALL unlocks all front panel buttons and knobs.

USBDevice:CONFigure

This command may be used to configure the rear USB port to be off or enabled as a USBTMC device. Users should be cautious using this command via the USBTMC interface as a change to the configuration of this interface from a USBTMC device will cause USBTMC communication to cease. It is intended to be used via the Ethernet interface to control the USB device interface.

Group      Miscellaneous
Syntax     USBDevice:CONFigure {DISabled|USBTmc}
           USBDevice:CONFigure?
Arguments  DISabled will disable the rear USB port.
           USBTmc enables the rear USB port.
Commands listed in alphabetical order

Examples

USBDEVICE:CONFIGURE DISABLED will disable the rear USB port.
USBDEVICE:CONFIGURE? might return :USBDEVICE:CONFIGURE USBTMC indicating the USB port is enabled.

VERBose

This command sets or queries the Verbose state that controls the length of keywords on query responses. Keywords can be both headers and arguments.

NOTE. This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk).

Group

Miscellaneous

Syntax

VERBose {<NR1>|OFF|ON}

Related Commands

HEADer

*LRN?

SET?

Arguments

<NR1> = 0 disables Verbose, any other value enables Verbose.
OFF sets the Verbose state to false, which returns minimum-length keywords for applicable setting queries.
ON sets the Verbose state to true, which returns full-length keywords for applicable setting queries.

A 0 returns minimum-length keywords for applicable setting queries, any other value returns full-length keywords.

Examples

VERBOSE ON sets the Verbose state to true and return the full length keyword for the applicable setting queries.
VERBOSE? might return :VERBOSE OFF, indicating that the Verbose state is set to false and return the minimum-length keywords for the applicable setting queries.

With :HEADER ON and :VERBOSE ON, the :ACQUIRE:MODE? query might return :ACQUIRE:MODE SAMPLE

With :HEADER ON and :VERBOSE OFF, the :ACQUIRE:MODE? query might return :ACQ:MOD SAM
With :HEADER OFF and :VERBOSE ON, the :ACQUIRE:MODE? query might return: SAMPLE

With :HEADER OFF and :VERBOSE OFF, the :ACQUIRE:MODE? query might return: SAM

**VERTical:DESKew:FROM:CUSTOMP**RAGation

This command sets or queries a target (FROM) delay that you can specify when the propagation delay of the target (FROM) probe used for deskew cannot be detected automatically.

**Group**  
Vertical

**Syntax**  
VERTical:DESKew:FROM:CUSTOMPRAgation <NR3>  
VERTical:DESKew:FROM:CUSTOMPRAgation?

**Arguments**  
<NR3> is a target (FROM) delay that you can specify when the propagation delay of the target (FROM) probe used for deskew cannot be detected automatically.

**Examples**  
VERTICAL:DESKew:FROM:CUSTOMPRAgATION 0.25e-9 sets the custom delay to 250 ps.

VERTICAL:DESKew:FROM:CUSTOMPRAgation? might return :VERTICAL:DESKew:FROM:CUSTOMPRAgation 0.0E+0 indicating the custom delay is 0.0 s.

**VERTical:DESKew:FROM:SOUrce**

This command sets or queries the source channel for performing channel-to-channel deskew adjustment. Sources can be any of the analog channels.

**Group**  
Vertical

**Syntax**  
VERTical:DESKew:FROM:SOURCE CH<x>  
VERTical:DESKew:FROM:SOURCE?

**Arguments**  
The vertical deskew source.
Examples

\texttt{VERTICAL:DESKEW:FROM:SOURCE CH1} sets channel 1 as the from source.

\texttt{VERTICAL:DESKEW:FROM:SOURCE?} might return

\texttt{:VERTICAL:DESKEW:FROM:SOURCE CH2} indicating channel 2 is the from source.

\section*{VERTical:DESKEW:STATIC (No Query Form)}

This command executes static deskew using the deskew settings.

\textbf{Group} Vertical

\textbf{Syntax} \texttt{VERTical:DESKEW:STATIC EXECute}

\textbf{Arguments} \texttt{EXECute} will execute static deskew using the deskew settings.

\textbf{Examples} \texttt{VERTICAL:DESKEW:STATIC EXECUTE} will execute static deskew using the deskew settings.

\section*{VERTical:DESKew:TO:CUSTOMPROPAGation}

This command sets or queries a target (TO) delay that can be specified by the user when the propagation delay of the target (TO) probe used for deskew cannot be detected automatically.

\textbf{Group} Vertical

\textbf{Syntax} \texttt{VERTical:DESKew:TO:CUSTOMPROPAGation <NR3>}

\textbf{Arguments} \texttt{<NR3>} is a target (TO) delay that can be specified by the user when the propagation delay of the target (TO) probe used for deskew cannot be detected automatically.

\textbf{Examples} \texttt{VERTICAL:DESKew:TO:CUSTOMPROPAGATION 0.25e-9} sets the custom propagation to 250 ps.

\texttt{VERTICAL:DESKew:TO:CUSTOMPROPAGATION?} might return

\texttt{:VERTICAL:DESKew:TO:CUSTOMPROPAGATION 0.0E+0} indicating the custom propagation is set to 0.0 ns.
VERTical:DESKEW:TO:SOUrce

This command sets or queries the target channel for performing channel-to-channel deskew adjustment. Target sources can be any of the live analog channels.

**Group**  
Vertical

**Syntax**  
VERTical:DESKEW:TO:SOURce CH<x>  
VERTical:DESKEW:TO:SOURce?

**Arguments**  
Arguments are the live analog channels.

**Examples**  
VERTICAL:DESKEW:TO:SOURCE CH4 sets the deskew to source is channel 4.  
VERTICAL:DESKEW:TO:SOURCE? might return :VERTICAL:DESKEW:TO:SOURCE CH2 indicating the deskew to source is channel 2.

VISual:AREA<x>:ASPEctratio

Sets or queries whether the aspect ratio of the specified Visual Trigger area is locked.

**Group**  
Trigger

**Syntax**  
VISual:AREA<x>:ASPEctratio {ON|OFF|<NR1>}  
VISual:AREA<x>:ASPEctratio?

**Arguments**  
Area<x> specifies the integer number of a Visual Trigger area.  
ON locks the aspect ratio of the specified Visual Trigger area. The aspect ratio is kept constant when the height or width of the area changes.  
OFF unlocks the aspect ratio of the specified Visual Trigger area.  
<NR1> is an integer number. 1 locks the aspect ratio of the specified Visual Trigger area, any other value unlocks the aspect ratio of the specified Visual Trigger area.

**Examples**  
VISual:AREA6:ASPECTratio ON locks the aspect ratio of area 6.  
VISual:AREA2:ASPECTratio? might return VISual:AREA2:ASPECTratio OFF, indicating that the aspect ratio of area 2 is not locked.
**VISual:AREA<x>:FLIP:HORizontal** (No Query Form)

Flips the specified Visual Trigger area horizontally around its center point.

**Group**  
Trigger

**Syntax**  
VISual:AREA<x>:FLIP:HORizontal

**Related Commands**  
VISual:AREA<x>:FLIP:VERTical

**Arguments**  
Area<x> specifies the integer number of a Visual Trigger area.

**Examples**  
VISual:AREA1:FLIP:HORizontal horizontally flips area 1.

**VISual:AREA<x>:FLIP:VERTical** (No Query Form)

Flips the specified Visual Trigger area vertically around its center point.

**Group**  
Trigger

**Syntax**  
VISual:AREA<x>:FLIP:VERTical

**Related Commands**  
VISual:AREA<x>:FLIP:HORizontal

**Arguments**  
Area<x> specifies the integer number of a Visual Trigger area.

**Examples**  
VISual:AREA8:FLIP:VERTical vertically flips area 8.

**VISual:AREA<x>:HEIGHT**

Sets or queries the height of the specified Visual Trigger area.

**Group**  
Trigger

**Syntax**  
VISual:AREA<x>:HEIGHT <NR3>  
VISual:AREA<x>:HEIGHT?
Arguments
Area<x> specifies the integer number of a Visual Trigger area.
<NR3> specifies the height of the Visual Trigger area in amplitude.

Examples
VISual:AREA51:HEIGHT 3.5 sets the height of area 51 to 3.5 (volts or Amps).
VISual:AREA4:HEIGHT? might return VISual:AREA4:HEIGHT 15, indicating that the height of area 4 is 15.

VISual:AREA<x>:HITType
Sets or queries the area hit logic true condition for the specified Visual Trigger area.

Group Trigger

Syntax
VISual:AREA<x>:HITType {IN|OUT|DONTcare}
VISual:AREA<x>:HITType?

Arguments
Area<x> specifies the integer number of a Visual Trigger area.
IN specifies that the waveform must intrude anywhere into the specified area to be true.
OUT specifies that the waveform must not intrude anywhere into the specified area to be true.
DONTcare sets the area to a don’t care state, causing the area to be ignored. This is useful when you are developing a Visual Trigger condition and need to change the hit logic type of an area while keeping the area.

Examples
VISual:AREA2:HITType OUT sets the area 2 hit type to OUT.
VISual:AREA5:HITType? might return VISual:AREA5:HITType IN, indicating that the waveform must intrude into area 5 to be true.

VISual:AREA<x>:RESET (No Query Form)
Sets the specified Visual Trigger area shape to a default-sized triangle.

Group Trigger

Syntax VISual:AREA<x>:RESET
Commands listed in alphabetical order

**Arguments**

Area<x> specifies the integer number of a Visual Trigger area.

**Examples**

VISual:AREA2:RESET changes area 2 to a default triangle shape.

---

**VISual:AREA<x>:ROTAtion**

Sets or queries the rotation angle of the specified Visual Trigger area.

**Group**

Trigger

**Syntax**

VISual:AREA<x>:ROTAtion <NR3>

VISual:AREA<x>:ROTAtion?

**Arguments**

Area<x> specifies the integer number of a Visual Trigger area.

<NR3> specifies the rotation angle of the Visual Trigger area, in positive degrees from 0 to 360. Zero degrees is referenced from when the area was created.

**Examples**

VISual:AREA2:ROTAtion 45 rotates Visual Trigger area 2 by 45 degrees.

VISual:AREA1:ROTAtion? might return VISual:AREA1:ROTAtion -60 indicating that area 4 rotation position is minus 60 degrees.

---

**VISual:AREA<x>:SHAPE**

Sets or queries the current shape of the area.

**Group**

Trigger

**Syntax**

VISual:AREA<x>:SHAPE {TRIAngle|RECTangle|TRApezoid|HEXAgon}

VISual:AREA<x>:SHAPE?

**Arguments**

Area<x> specifies the integer number of a Visual Trigger area.

TRIAngle sets the specified area to a triangular shape (three vertices). If the area does not exist, the instrument creates a new triangular area with the specified area number.

RECTangle sets the specified area to a rectangular shape (four vertices, right angles at each corner). If the area does not exist, the instrument creates a new triangular area with the specified area number.
TRAPEZOID sets the specified area to a trapezoidal shape (four vertices, two parallel sides). If the area does not exist, the instrument creates a new triangular area with the specified area number.

HEXAGON sets the specified area to a hexagonal shape (six vertices). If the area does not exist, the instrument creates a new hexagonal area with the specified area number.

Returns

CUSTOM indicates that the shape is a custom-created shape, or the positions of one or more vertices of a standard shape have been changed.

Examples

VISual:AREA1:SHAPE Hexagon sets area 1 to be a hexagonal shape, or creates a new area 1 hexagonal shape if the specified area does not exist.

VISual:AREA2:SHAPE? might return VISual:AREA2:SHAPE CUSTOM, indicating that area 2 is a custom shape.

VISual:AREA<x>:SOUrce

Sets or queries the signal source for the specified Visual Trigger area. The source can only be an analog channel.

Group

Trigger

Syntax

VISual:AREA<x>:SOUrce {CH1|CH2|CH3|CH4|CH5|CH6|CH7|CH8}
VISual:AREA<x>:SOUrce?

Arguments

Area<x> specifies the integer number of a Visual Trigger area.

CH1 through CH8 sets the source channel number for the specified area.

Examples

VISual:AREA1:SOUrce CH3 sets the source channel number for area 1 to Channel 3.

VISual:AREA6:SOUrce? might return VISual:AREA6:SOUrce CH8, indicating that the source for area 6 is Channel 8.

VISual:AREA<x>:VERTICES

Sets or queries the X and Y vertex coordinate values for all vertices of the specified Visual Trigger area. Vertex values must be set in pairs.
Group Trigger

Syntax

**VISual:AREA<x>:VERTICES** “<NR3>, <NR3>, <NR3>, <NR3>, <NR3>,
<NR3> [,<NR3>, <NR3>, ...]”

**VISual:AREA<x>:VERTICES?**

Arguments

Area<x> specifies the integer number of a Visual Trigger area.<br>
<NR3>, <NR3> specifies the X, Y coordinate pair of each vertex in an area. The first value is X (time) and the second value is Y (amplitude). There must be a minimum of three vertex pairs to create an area. If the specified area exists, the area is changed to the shape specified in the arguments. If the specified area does not exist, a new area is created and assigned the specified vertices.

Examples

**VISual:AREA3:VERTICES**

"400E-9,1,400E-9,0.5,600E-9,0.5,600E-9,1" sets or creates area 3 as a rectangle that is 200 ns wide by 500 mV high.

**VISual:AREA3:VERTICES?** might return **VISual:AREA3:VERTICES**

"171.1E-9,-141.7E-3,321.1E-9,-141.7E-3,283.6E-9,1.708,
208.6E-9,1.708", indicating that area 3 is defined as a trapezoid that is 150 ns wide by 1.85 V high.

**VISual:AREA<x>:WIDTH**

Sets or queries the width of the specified Visual Trigger area.

Group Trigger

Syntax

**VISual:AREA<x>:WIDTH** <NR3>

**VISual:AREA<x>:WIDTH?**

Arguments

Area<x> specifies the integer number of a Visual Trigger area.<br>
<NR3> specifies the width of the Visual Trigger area in seconds.

Examples

**VISual:AREA5:WIDTH** 0.0000045 sets the width of area 5 to 4.5 μs.

**VISual:AREA3:WIDTH?** might return **VISual:AREA3:WIDTH** 0.000016, indicating that the width of area 3 is 16 μs.
VISual:AREA<x>:XPOSition

Sets or queries the horizontal (X-axis) center position of the specified Visual Trigger area.

Group       Trigger

Syntax       VISual:AREA<x>:XPOSition <NR3>
             VISual:AREA<x>:XPOSition?

Arguments    Area<x> specifies the integer number of a Visual Trigger area.
             <NR3> specifies the horizontal position of the center of the Visual Trigger area, in seconds.

Examples     VISual:AREA1:XPOSition -6.9e-6 sets the horizontal center of area 1 to be -6.9 μs from the trigger point.
             VISual:AREA3:XPOSition? might return VISual:AREA3:XPOSition 8.0e-6, indicating that the center of area 3 is 8 μs from the trigger point.

VISual:AREA<x>:YPOSition

Sets or queries the vertical (Y-axis) center position of the specified Visual Trigger area.

Group       Trigger

Syntax       VISual:AREA<x>:YPOSition <NR3>
             VISual:AREA<x>:YPOSition?

Arguments    Area<x> specifies the integer number of a Visual Trigger area.
             <NR3> specifies the vertical position of the center of the Visual Trigger area, in amplitude (volts, amps).

Examples     VISual:AREA1:YPOSition 0.5 sets the vertical center of area 1 to be 500 mV.
             VISual:AREA3:YPOSition? might return VISual:AREA3:YPOSition 2, indicating that the vertical center of area 3 is 2 volts.
**VISual:DELETEALL (No Query Form)**

Deletes all Visual Trigger areas.

**Group**

Trigger

**Syntax**

VISual:DELETEALL

**Examples**

VISUAL:DELETEALL deletes all Visual Trigger areas.

**VISual:ENAble**

Sets or queries the status (on or off) of the Visual Trigger mode.

**Group**

Trigger

**Syntax**

VISual:ENAble {ON|OFF|<NR1>}

VISual:ENAble?

**Arguments**

ON enables the Visual Trigger mode.

OFF disables the Visual Trigger mode.

<NR1> is an integer number. 0 turns off the Visual Trigger mode; any other value enables Visual Trigger mode.

**Examples**

VISual:ENAble ON enables the Visual Trigger function.

VISual:ENAble? might return VISual:ENAble OFF, indicating that Visual Trigger mode is not enabled.

**VISual:EQUation**

Sets or queries the Visual Trigger area combination logic equation.

**Group**

Trigger

**Syntax**

VISual:EQUation <Qstring>

VISual:EQUation?
Commands listed in alphabetical order

Arguments

<qstring> defines the Visual Trigger area combination logic equation. The equation is made up of area names \( A<x> \) combined with logic operators AND, OR, or XOR (\&, |, ^). It may also contain parentheses for grouping. The equation must be true to have a valid Visual Trigger event and display a waveform. Each area’s true state depends on the area’s condition setting (In, Out or Don’t Care).

Examples

VISual:EQUation "(A1 & A2) | A3" sets the combined area logic such that both areas 1 and 2 must be true, or area 3 must be true, to have a valid Visual Trigger event and display a waveform.

VISual:EQUation? might return VISual:EQUation "A1 & A2 & A3 & A4", indicating that the area combination logic equation requires that areas 1 through 4 must be true to have a valid Visual Trigger event.

VISual:SHOWAREas

Shows or hides all Visual Trigger areas.

Group

Trigger

Syntax

VISual:SHOWAREas {ON|OFF|<NR1>}

VISual:SHOWAREas?

Arguments

ON shows all Visual Trigger areas.

OFF hides all Visual Trigger areas.

<NR1> is an integer number. 0 hides all Visual Trigger areas; any other value shows all Visual Trigger areas.

Examples

VISual:SHOWAREas OFF hides all Visual Trigger areas.

VISual:SHOWAREas? might return VISual:SHOWAREas ON, indicating that Visual Trigger areas are displayed on the screen.

VISual:SHOWCRiteria

Sets or queries display of the area names and hit criteria for all visual trigger areas.

Group

Trigger
Commands listed in alphabetical order

**Syntax**

VISual:SHOWCriteria {ON|OFF|<NR1>}

VISual:SHOWCriteria?

**Arguments**

ON enables display of the area name and hit criteria labels (In, Out, Don’t care icons) of all Visual Trigger areas.

OFF hides the area name and hit criteria labels (In, Out, Don’t care icons) of all Visual Trigger areas.

<NR1> is an integer number. 0 hides the area name and hit criteria of all Visual Trigger areas, any other value enables displaying the area name and hit criteria of all Visual Trigger areas.

**Examples**

VISual:SHOWCriteria OFF hides the name and hit criteria labels of all Visual Trigger areas.

VISual:SHOWCriteria? might return VISual:SHOWCriteria ON, indicating that the name and hit criteria labels of all Visual Trigger areas are displayed.

**VISual:SHOWEquation**

Shows or hides the Visual Trigger area combination logic equation.

**Group**

Trigger

**Syntax**

VISual:SHOWEquation {ON|OFF|<NR1>}

VISual:SHOWEquation?

**Arguments**

ON shows the Visual Trigger area combination logic equation.

OFF hides the Visual Trigger area combination logic equation.

<NR1> is an integer number. 0 hides the area combination logic equation, any other value displays the area combination logic equation.

**Examples**

VISual:SHOWEquation ON shows the Visual Trigger area combination logic equation.

VISual:SHOWEquation? might return VISual:SHOWEquation OFF, indicating that the Visual Trigger area combination logic equation is not displayed on the screen.
**WAI (No Query Form)**

The *WAI (Wait) command (no query form) prevents the instrument from executing further commands or queries until all pending commands that generate an OPC message are complete. This command allows you to synchronize the operation of the instrument with your application program. For more information, refer to Synchronization Methods.

<table>
<thead>
<tr>
<th>Group</th>
<th>Status and Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>*WAI</td>
</tr>
</tbody>
</table>
| Related Commands | BUSY?  
               | *OPC                                                  |
| Examples    | *WAI prevents the instrument from executing any further commands or queries until all pending commands that generate an OPC message are complete. |

**WAVFrm? (Query Only)**

This query-only command provides the Tektronix standard waveform query which returns the waveform preamble followed by the waveform data for the source specified by :DATa:SOURce using the :DATa settings for encoding, width, and so forth.

<table>
<thead>
<tr>
<th>Group</th>
<th>Waveform Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>WAVFrm?</td>
</tr>
</tbody>
</table>
| Related Commands | CURVe  
               | DATa:SOURce                                           |
|             | WFMOOutpre?                                           |
| Examples    | WAVFrm? might return the waveform data as: WFMOUTPRE:BIT_NR 8;BN_FMT R;BYT_NR 1;BYT_OR MSB;ENCDG ASC;NR_PT 500;PT_FMT Y;PT_ORDER LINEAR;PT_OFF 0;XINC 400.0000E-12;XZERO 0.0000;XUNIT "s";YMULT 4.0000E-3;YOFF 0.0000;YZERO 0.0000;YUNIT "V";WFID "Ch1,DC coupling, 100.0mV/div, 200.0ns/div, 5000 points,Samp1 mode". |
WFMOOutpre? (Query Only)

This query-only command queries the waveform formatting data for the waveform specified by the DATa:SOURce command. The preamble components are considered to be of two types; formatting and interpretation. The formatting components are: ENCdg, BN_FMT, BYT_Or, BYT_Nr, BIT_Nr. The interpretation components are derived from the DATa:SOURce specified waveform.

Group: Waveform Transfer

Syntax: WFMOOutpre?

Examples: WFMOOutpre? might return the waveform formatting data as:

```
:WFMOOutpre:BYT_NR 2;BIT_NR 16;ENCdg BINARY;BN_FMT RI;BYT_Or MSB;WFID "Ch1, DC coupling, 200.0mV/div, 10.00us/div, 1250 points, Sample mode";NR_PT 1000;PT_FMT Y;XUNIT "s";XINCR 80.0000E-9;XZERO 0.0000;PT_OFF 625;YUNIT "V";YMULT 31.2500E-6;YOFF 0.0000;YZERO 0.0000;NR_FR 3.
```

WFMOOutpre:ASC_Fmt? (Query Only)

This query returns the format for ASCII data transferred from the instrument. No command form is provided as the format is determined by the data source type. Some waveforms are normalized vector data where the data points are 8-byte doubles in floating point format whereas other formats are 1-byte or 2-byte integers.

Group: Waveform Transfer

Syntax: WFMOOutpre:ASC_Fmt?

Related Commands: DATa:SOURce

WFMOOutpre:BN_FMT

WFMOOutpre:ENCdg

Returns: FP represents floating point ASCII data. The waveforms are normalized vector data where the data points are 8-byte doubles in floating point format.
INTEGER represents signed integer ASCII data. The waveform data are 1-byte or 2-byte integers.

Examples

WFMOutpre:ASC_Fmt? might return :WFMOutpre:ASC_FMT INTEGER indicating the ASCII format is integer.

WFMOutpre:BIT_Nr

This command sets and queries the number of bits per waveform point that outgoing waveforms contain, as specified by the DATa:SOURce command. Note that values will be constrained according to the underlying waveform data. This specification is only meaningful when WFMOutpre:ENCdg is set to BIN and WFMOutpre:BN_Fmt is set to either RI or RP.

Group Waveform Transfer

Syntax

WFMOutpre:BIT_Nr <NR1>
WFMOutpre:BIT_Nr?

Related Commands

DATa:SOURce
WFMOutpre:BN_Fmt
WFMOutpre:ENCdg

Arguments

<NR1> number of bits per data point can be 8 or 16.

Examples

WFMOutpre:BIT_Nr 16 sets the number of bits per waveform point to 16 for incoming RI and RP binary format data.

WFMOutpre:BIT_Nr? might return :WFMOutpre:BIT_Nr 8, indicating that outgoing RI or RP binary format data uses 8 bits per waveform point.

WFMOutpre:BN_Fmt

This command sets or queries the format of binary data for outgoing waveforms specified by the DATa:SOURce command.

Group Waveform Transfer
Commands listed in alphabetical order

**Syntax**

WFMOutpre:BN_FMT {RI|RP|FP}
WFMOutpre:BN_FMT?

**Related Commands**

DATa:SOUrce

**Arguments**

RI specifies signed integer data point representation.
RP specifies positive integer data point representation.
FP specifies floating point representation.

**Examples**

WFMOUTPRE:BN_FMT FP specifies that outgoing waveform data will be in single-precision binary floating point format.
WFMOUTPRE:BN_FMT? might return :WFMOUTPRE:BN_FMT RI, indicating that the outgoing waveform data is currently in signed integer format.

**WFMOutpre:BYT_Nr**

This command sets or queries the binary field data width (bytes per point) for the waveform specified by the DATa:SOUrce command. Note that values will be constrained according to the underlying waveform data. This specification is only meaningful when WFMOutpre:ENCdg is set to BIN, and WFMOutpre:BN_FMT is set to either RI or RP.

**Group**

Waveform Transfer

**Syntax**

WFMOutpre:BYT_Nr <NR1>
WFMOutpre:BYT_Nr?

**Related Commands**

DATa:SOUrce
WFMOutpre:BN_FMT
WFMOutpre:ENCdg

**Arguments**

<NR1> is the number of bytes per data point and can be 1, 2 or 8. A value of 1 or 2 bytes per waveform point indicates channel data, 8 bytes per waveform point indicate pixel map (fast acquisition) data.

**Examples**

WFMOUTPRE:BYT_NR 1 sets the number of bytes per outgoing waveform data point to 1, which is the default setting.
WFMOutpre:BYT_Or

This command sets or queries which byte of binary waveform data is transmitted first, during a waveform data transfer, when data points require more than one byte. This specification only has meaning when WFMOutpre:ENCdg is set to BIN.

**Group**  
Waveform Transfer

**Syntax**  
WFMOutpre:BYT_Or {LSB|MSB}  
WFMOutpre:BYT_Or?

**Related Commands**  
WFMOutpre:ENCdg

**Arguments**  
LSB specifies that the least significant byte will be transmitted first.  
MSB specifies that the most significant byte will be transmitted first.

**Examples**  
WFMOUTPRE:BYT_OR MSB sets the most significant outgoing byte of waveform data to be transmitted first.

WFMOUTPRE:BYT_OR? might return :WFMOUTPRE:BYT_OR LSB, indicating that the least significant data byte will be transmitted first.

WFMOutpre:DOMain? (Query Only)

This query returns the domain of the outgoing waveform.

**Group**  
Waveform Transfer

**Syntax**  
WFMOutpre:DOMain?

**Related Commands**  
DATa:SOUrce  
WFMOutpre:BN_Fmt  
WFMOutpre:ENCdg
 Commands listed in alphabetical order

**Returns**

Returns the domain of the outgoing waveform.

**Examples**

`WFMOUTPRE:DOMAIN?` might return `:WFMOUTPRE:DOMAIN TIME`, indicating that the outgoing waveform is time domain trace.

**WFMOutpre:ENCdg**

This command sets or queries the type of encoding for outgoing waveforms.

**Group**

Waveform Transfer

**Syntax**

```
WFMOutpre:ENCdg {ASCii|BINary}
WFMOutpre:ENCdg?
```

**Related Commands**

- `DATa:ENCdg`
- `WFMOutpre.BYT_Nr`
- `WFMOutpre.BYT_Or`
- `WFMOutpre.BIT_Nr`
- `WFMOutpre.BN_Fmt`

**Arguments**

- **ASCii** specifies that the outgoing data is to be in ASCII format. Waveforms internally stored as integers will be sent as `<NR1>` numbers, while those stored as floating point will be sent as `<NR3>` numbers.
- **BINary** specifies that outgoing data is to be in a binary format whose further specification is determined by `WFMOutpre.BYT_Nr`, `WFMOutpre.BIT_Nr`, `WFMOutpre.BN_FMT` and `WFMOutpre.BYT_Or`.

**Examples**

- `WFMOUTPRE:ENCdg ASCii` specifies that the outgoing waveform data will be sent in ASCII format.
- `WFMOUTPRE:ENCdg?` might return `:WFMOUTPRE:ENCdg BINary`, indicating that outgoing waveform data will be sent in binary format.

**WFMOutpre:NRPt? (Query Only)**

This query-only command returns the number of points for the `DATa:SOURce` waveform that will be transmitted in response to a `CURVe?` query.
Group     Waveform Transfer
Syntax    WFMOutpre:NR_Pt?

Related Commands    CURVe
                     DATa
                     DATa:STARt
                     DATa:STOP
                     SAVE:WAVEform
                     SAVE:ON:WAVEform:FILEFormat

Examples    WFMOutpre:NR_Pt? might return :WFMOutpre:NR_Pt 5000, indicating that there are 5000 data points to be sent.

WFMOOutpre:PT_Fmt? (Query Only)

This query-only command returns the point format for the waveform specified by the DATa:SOURce command. The format specifies a set of equations describing how the scale factors in the preamble are used to give meaning to the CURVe data points.

An error is reported if the DATa:SOURce waveform does not exist.

Group     Waveform Transfer
Syntax    WFMOOutpre:PT_Fmt?

Related Commands    CURVe
                     DATa:SOURce

Examples    WFMOOutpre:PT_Fmt? might return :WFMOOutpre:PT_Fmt ENV, indicating that the waveform data is a series of min-max pairs.

WFMOOutpre:PT_Off? (Query Only)

This query-only command returns the trigger point relative to DATa:STARt for the waveform specified by the DATa:SOURce command.
NOTE. This returned value is the point immediately following the actual trigger.

<table>
<thead>
<tr>
<th>Group</th>
<th>Waveform Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>WFMOutpre:PT_off?</td>
</tr>
<tr>
<td>Related Commands</td>
<td>DATa:SOURce</td>
</tr>
<tr>
<td></td>
<td>DATa:STARt</td>
</tr>
</tbody>
</table>
|             | WFMOutpre:XZEr?

Examples

WFMOutpre:PT_OFF? might return :WFMOUTPRE:PT_OFF 251 specifying that the trigger actually occurred between points 250 and 251.

WFMOutpre:WFId? (Query Only)

This query-only command returns a string describing several aspects of the acquisition parameters for the waveform specified by the DATa:SOURce command. An error is reported if the DATa:SOURce waveform does not exist.

<table>
<thead>
<tr>
<th>Group</th>
<th>Waveform Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>WFMOutpre:WFId?</td>
</tr>
<tr>
<td>Related Commands</td>
<td>DATa:SOURce</td>
</tr>
<tr>
<td>Returns</td>
<td>&lt;QString&gt; contains the following comma-separated fields documented in the following tables:</td>
</tr>
</tbody>
</table>

Table 2-47: Waveform Suffixes

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>The source identification string as it appears in the front panel scale factor readouts.</td>
<td>&quot;CH1–8&quot; &quot;Math&lt;x&gt;&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;Ref&lt;x&gt;&quot;</td>
</tr>
<tr>
<td>Coupling</td>
<td>A string describing the vertical coupling of the waveform (the Source1 waveform in the case of Dual Waveform Math).</td>
<td>&quot;AC coupling&quot; &quot;DC coupling&quot; &quot;GND coupling&quot;</td>
</tr>
</tbody>
</table>
Table 2-47: Waveform Suffixes (cont.)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vert Scale</td>
<td>A string containing the vertical scale factor of the unzoomed waveform. The numeric portion will always be four digits. The examples cover all known internal units.</td>
<td>&quot;100.0 mV/div&quot; &quot;20.00 dB/div&quot; &quot;45.00 deg/div&quot; &quot;785.4 mrad/div&quot; &quot;500.0 μVs/div&quot; &quot;10.00 kV/s/div&quot; &quot;200.0 mV/div&quot; &quot;50.00 unk/div&quot;</td>
</tr>
<tr>
<td>Horiz Scale</td>
<td>A string containing the horizontal scale factor of the unzoomed waveform. The numeric portion will always be four digits. The examples cover all known internal units.</td>
<td>&quot;100 ms/div&quot; &quot;10.00 kHz/div&quot; &quot;50.00 c/div&quot;</td>
</tr>
<tr>
<td>Record Length</td>
<td>A string containing the number of waveform points available in the entire record. The numeric portion is given as an integer.</td>
<td>&quot;500 points&quot; &quot;500000 points&quot;</td>
</tr>
<tr>
<td>Acquisition Mode</td>
<td>A string describing the mode used to acquire the waveform.</td>
<td>&quot;Sample mode&quot; &quot;Pk Detect mode&quot; &quot;Hi Res mode&quot; &quot;Envelope mode&quot; &quot;Average mode&quot;</td>
</tr>
</tbody>
</table>

Examples

WFMOUTPRE:WFID? might return :WFMOUTPRE:WFID "Ch1, DC coupling,100.0mVolts/div,500.0µs/div,500 points, Hi Res mode".

WFMOutpre:XINcr? (Query Only)

This query-only command returns the horizontal point spacing in units of WFMOutpre:XUNit for the waveform specified by the DATa:SOURce command. This value corresponds to the sampling interval.

An error is reported if the DATa:SOURce waveform does not exist.

Group       Waveform Transfer

Syntax       WFMOutpre:XINcr?

Related Commands  DATa:SOURce  WFMOutpre:XUNit?
Commands listed in alphabetical order

Examples

WFMOUTPRE:XINCR? might return WFMOUTPRE:XINCR 1.0000E-6, indicating that the horizontal sampling interval is 10 µs/point (500 µs/div).

WFMOutpre:XUNit? (Query Only)

This query-only command returns the horizontal units for the waveform specified by the DATa:SOUrce command. An error is reported if the DATa:SOUrce waveform does not exist.

Group Waveform Transfer

Syntax WFMOutpre:XUNit?

Related Commands DATa:SOUrce

Examples WFMOUTPRE:XUNIT? might return WFMOUTPRE:XUNIT "HZ", indicating that the horizontal units for the waveform are in Hertz.

WFMOutpre:XZEr0? (Query Only)

This query-only command returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa:SOUrce command. This value is in units of WFMOutpre:XUNit.

An error is reported if the DATa:SOUrce waveform does not exist.

NOTE: During steady state operation, when all control changes have settled and triggers are arriving on a regular basis, this is the only part of the preamble that changes on each acquisition.

Group Waveform Transfer

Syntax WFMOutpre:XZEr0?

Related Commands DATa:SOUrce

WFMOutpre:XUNit?
Examples

WFMOOutpre:XZERO? might return WFMOOutpre:XZERO 5.6300E-9, indicating that the trigger actually occurred 5.63 ns before the trigger sample.

WFMOOutpre:YMUlt? (Query Only)

This query-only command returns the vertical scale factor per digitizing level in units specified by WFMOOutpre:YNUit for the waveform specified by the DATa:SOUrce command. For those formats in which WFMOOutpre:BYT_NR is important (all non-floating point formats), WFMOOutpre:YMUlt? must take the location of the binary point implied by BYT_NR into consideration.

An error is reported if the DATa:SOUrce waveform does not exist.

Group Waveform Transfer

Syntax WFMOOutpre:YMUlt?

Related Commands DATa:SOUrce

Examples WFMOOUTPRE:YMULT? might return WFMOOUTPRE:YMULT 4.0000E-3, indicating that the vertical scale for the corresponding waveform is 100 mV/div.

WFMOOutpre:YOFf? (Query Only)

This query-only command returns the vertical offset of the source specified by DATa:SOUrce. For this instrument family, the value returned is always 0.0 as the offset is combined with the :YZEro value.

An error is reported if the DATa:SOUrce waveform does not exist.

Group Waveform Transfer

Syntax WFMOOutpre:YOFf?

Related Commands DATa:SOUrce

WFMOOutpre:BYT_NR
Commands listed in alphabetical order

Examples  WFMOUTPRE:YOFF? might return WFMOUTPRE:YOFF -50.0000E+0, indicating that the position indicator for the waveform was 50 digitizing levels (2 divisions) below center screen.

**WFMOutpre:YUNit? (Query Only)**

This query-only command returns the vertical units for the waveform specified by the DATa:SOURce command.

An error is reported if the DATa:SOURce waveform does not exist.

Group  Waveform Transfer

Syntax  WFMOutpre:YUNit?

Related Commands  DATa:SOURce

Examples  WFMOUTPRE:YUNIT? might return WFMOUTPRE:YUNIT "dB", indicating that the vertical units for the waveform are measured in decibels.

**WFMOutpre:YZEro? (Query Only)**

This query-only command returns the combined vertical position and offset for the source waveform specified by DATa:SOURce. This represents a departure from previous instruments where the :YZEro value represented the vertical position in vertical units and the :YOFF value represented the vertical offset in digitizing levels. For this instrument family, the value of :YOFF is always 0.0.

An error is reported if the DATa:SOURce waveform does not exist.

Group  Waveform Transfer

Syntax  WFMOutpre:YZEro?

Related Commands  DATa:SOURce

Examples  WFMOUTPRE:YZERO? might return WFMOUTPRE:YZERO -100.0000E-3, indicating that vertical offset is set to -100 mV.
Status and Events

The oscilloscope provides a status and event reporting system for the Ethernet and USB interfaces. This system informs you of certain significant events that occur within the oscilloscope.

The oscilloscope status handling system consists of five 8-bit registers and two queues for each interface. The remaining Status subtopics describe these registers and components. They also explain how the event handling system operates.

Registers

Overview

The registers in the event handling system fall into two functional groups:

- **Status Registers** contain information about the status of the oscilloscope. They include the Standard Event Status Register (SESR).
- **Enable Registers** determine whether selected types of events are reported to the Status Registers and the Event Queue. They include the Device Event Status Enable Register (DESER), the Event Status Enable Register (ESER), and the Service Request Enable Register (SRER).

Status Registers

The Standard Event Status Register (SESR) and the Status Byte Register (SBR) record certain types of events that may occur while the oscilloscope is in use. IEEE Std 488.2-1987 defines these registers.

Each bit in a Status Register records a particular type of event, such as an execution error or message available. When an event of a given type occurs, the oscilloscope sets the bit that represents that type of event to a value of one. (You can disable bits so that they ignore events and remain at zero. See Enable Registers). Reading the status registers tells you what types of events have occurred.

**The Standard Event Status Register (SESR)**. The SESR records eight types of events that can occur within the oscilloscope. Use the *ESR? query to read the SESR register. Reading the register clears the bits of the register so that the register can accumulate information about new events.

**NOTE.** TekVISA applications use SESR bit 6 to respond to any of several events, including some front panel actions.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>PON</td>
</tr>
<tr>
<td>6</td>
<td>URQ</td>
</tr>
<tr>
<td>5</td>
<td>CME</td>
</tr>
<tr>
<td>4</td>
<td>EXE</td>
</tr>
<tr>
<td>3</td>
<td>DDE</td>
</tr>
<tr>
<td>2</td>
<td>QYE</td>
</tr>
<tr>
<td>1</td>
<td>RQC</td>
</tr>
<tr>
<td>0</td>
<td>OPC</td>
</tr>
</tbody>
</table>

Figure 3-1: The Standard Event Status Register (SESR)
Table 3-1: SESR Bit Functions

<table>
<thead>
<tr>
<th>Bit</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 (MSB)</td>
<td>PON</td>
</tr>
<tr>
<td>6</td>
<td>URQ</td>
</tr>
<tr>
<td>5</td>
<td>CME</td>
</tr>
<tr>
<td>4</td>
<td>EXE</td>
</tr>
<tr>
<td>3</td>
<td>DDE</td>
</tr>
<tr>
<td>2</td>
<td>QYE</td>
</tr>
<tr>
<td>1</td>
<td>RQC</td>
</tr>
<tr>
<td>0 (LSB)</td>
<td>OPC</td>
</tr>
</tbody>
</table>

The Status Byte Register (SBR). Records whether output is available in the Output Queue, whether the oscilloscope requests service, and whether the SESR has recorded any events.

Use a Serial Poll or the *STB? query to read the contents of the SBR. The bits in the SBR are set and cleared depending on the contents of the SESR, the Event Status Enable Register (ESER), and the Output Queue. When you use a Serial Poll to obtain the SBR, bit 6 is the RQS bit. When you use the *STB? query to obtain the SBR, bit 6 is the MSS bit. Reading the SBR does not clear the bits.

![Figure 3-2: The Status Byte Register (SBR)](image)

Table 3-2: SBR Bit Functions

<table>
<thead>
<tr>
<th>Bit</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 (MSB)</td>
<td>Not used.</td>
</tr>
<tr>
<td>6</td>
<td>RQS</td>
</tr>
<tr>
<td>6</td>
<td>MSS</td>
</tr>
<tr>
<td>5</td>
<td>ESB</td>
</tr>
</tbody>
</table>
Table 3-2: SBR Bit Functions (cont.)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>MAV Message Available. Shows that output is available in the Output Queue.</td>
</tr>
<tr>
<td>3</td>
<td>Not used.</td>
</tr>
<tr>
<td>2</td>
<td>Not used.</td>
</tr>
<tr>
<td>1–0</td>
<td>Not used.</td>
</tr>
</tbody>
</table>

Enable Registers

DESER, ESER, and SRER allow you to select which events are reported to the Status Registers and the Event Queue. Each Enable Register acts as a filter to a Status Register (the DESER also acts as a filter to the Event Queue) and can prevent information from being recorded in the register or queue.

Each bit in an Enable Register corresponds to a bit in the Status Register it controls. In order for an event to be reported to a bit in the Status Register, the corresponding bit in the Enable Register must be set to one. If the bit in the Enable Register is set to zero, the event is not recorded.

Various commands set the bits in the Enable Registers. The Enable Registers and the commands used to set them are described below.

The Device Event Status Enable Register (DESER). This register controls which types of events are reported to the SESR and the Event Queue. The bits in the DESER correspond to those in the SESR.

Use the DESE command to enable and disable the bits in the DESER. Use the DESE? query to read the DESER.

![Figure 3-3: The Device Event Status Enable Register (DESER)](image)

The Event Status Enable Register (ESER). This register controls which types of events are summarized by the Event Status Bit (ESB) in the SBR. Use the *ESE command to set the bits in the ESER. Use the *ESE? query to read it.

![Figure 3-4: The Event Status Enable Register (ESER)](image)

The Service Request Enable Register (SRER). This register controls which bits in the SBR generate a Service Request and are summarized by the Master Status Summary (MSS) bit.
Use the *SRE command to set the SRER. Use the *SRE? query to read the register. The RQS bit remains set to one until either the Status Byte Register is read with a Serial Poll or the MSS bit changes back to a zero.

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
</table>

Figure 3-5: The Service Request Enable Register (SRER)

**PSC Command**

The *PSC command controls the Enable Registers contents at power-on. Sending *PSC 1 sets the Enable Registers at power on as follows:

- DESER 255 (equivalent to a DESe 255 command)
- ESER 0 (equivalent to an *ESE 0 command)
- SRER 0 (equivalent to an *SRE 0 command)

Sending *PSC 0 lets the Enable Registers maintain their values in nonvolatile memory through a power cycle.

**NOTE.** To enable the PON (Power On) event to generate a Service Request, send *PSC 0, use the DESe and *ESE commands to enable PON in the DESER and ESER, and use the *SRE command to enable bit 5 in the SRER. Subsequent power-on cycles will generate a Service Request.

**Queues**

The *PSC command controls the Enable Registers contents at power-on. Sending *PSC 1 sets the Enable Registers at power on as follows:

**Output Queue**

The oscilloscope stores query responses in the Output Queue and empties this queue each time it receives a new command or query message after an <EOM>. The controller must read a query response before it sends the next command (or query) or it will lose responses to earlier queries.

**CAUTION.** When a controller sends a query, an <EOM>, and a second query, the oscilloscope normally clears the first response and outputs the second while reporting a Query Error (QYE bit in the ESER) to indicate the lost response. A fast controller, however, may receive a part or all of the first response as well. To avoid this situation, the controller should always read the response immediately after sending any terminated query message or send a DCL (Device Clear) before sending the second query.
**Event Queue**

The Event Queue stores detailed information on up to 33 events. If than 32 events stack up in the Event Queue, the 32nd event is replaced by event code 350, "Queue Overflow."

Read the Event Queue with the EVENT? query (which returns only the event number), with the EVMSG? query (which returns the event number and a text description of the event), or with the ALLEV? query (which returns all the event numbers along with a description of the event). Reading an event removes it from the queue.

Before reading an event from the Event Queue, you must use the *ESR? query to read the summary of the event from the SESR. This makes the events summarized by the *ESR? read available to the EVENT? and EVMSG? queries, and empties the SESR.

Reading the SESR erases any events that were summarized by previous *ESR? reads but not read from the Event Queue. Events that follow an *ESR? read are put in the Event Queue but are not available until *ESR? is used again.

**Event Handling Sequence**

The following figure shows how to use the status and event handling system. In the explanation that follows, numbers in parentheses refer to numbers in the figure.
When an event occurs, a signal is sent to the DESER (1). If that type of event is enabled in the DESER (that is, if the bit for that event type is set to 1), the appropriate bit in the SESR is set to one, and the event is recorded in the Event Queue (2). If the corresponding bit in the ESER is also enabled (3), then the ESB bit in the SBR is set to one (4).

When output is sent to the Output Queue, the MAV bit in the SBR is set to one (5).

When a bit in the SBR is set to one and the corresponding bit in the SRER is enabled (6), the MSS bit in the SBR is set to one and a service request is generated (7).
Synchronization Methods

Overview

Although most commands are completed almost immediately after being received by the oscilloscope, some commands start a process that requires time. For example, once a single sequence acquisition command is executed, depending upon the applied signals and trigger settings, it may take an extended period of time before the acquisition is complete. Rather than remain idle while the operation is in progress, the oscilloscope will continue processing other commands. This means that some operations will not be completed in the order that they were sent. Furthermore, sometimes the result of an operation depends upon the result of an earlier operation. A first operation must complete before the next one is processed.

In order to handle these situations, the oscilloscope status and event reporting system allows you to synchronize the operation of the oscilloscope with your application program, using the Operation Complete function. Note, however, that only some operations are able to take advantage of this function; a table is provided below of commands that support this.

The following commands are used to synchronize the oscilloscope functions using Operation Complete. See examples of how to use these commands later on in this section:

*OPC — sending the *OPC command will set bit 0 of the SESR (Standard Events Status Register). The bit will only be set high when all pending operations that generate an OPC message have finished execution. (The SESR is queried using *ESR?) The *OPC? query form returns 1 only when all operations have completed, or a device clear is received.

*WAI — prevents the oscilloscope from executing further commands or queries until all pending commands that generate an OPC message are complete.

BUSY? — returns the status of operations: 1 means there are pending operations, 0 means that all pending operations are complete.

NOTE. Some OPC operations may require an extended period of time to complete or may never complete. For example, a single sequence acquisition may never complete when no trigger event occurs. You should be aware of these conditions and tailor your program accordingly by:

— setting the timeout sufficiently for the anticipated maximum time for the operation and

— handle a timeout appropriately by querying the SESR (*ESR?) and event queue (EVMsg? or ALLEv?).
NOTE. The *OPC command form can also be used to cause an SRQ to be generated upon completion of all pending operations. This requires that the ESB (Event Status Bit, bit 5) of the SRER (Service Request Enable Register) is set, and the OPC bit (bit 0) of the DESR (Device Event Status Enable Register) and the ESER (Event Status Enable Register) are set. (The SRER is set/queried using *SRE. The DESR is set/queried using DESE. The ESER is set/queried using *ESE.)

Only a subset of oscilloscope operations support the Operation Complete function (OPC).

Table 3-3: Oscilloscope operations that can generate OPC

<table>
<thead>
<tr>
<th>Command</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQuire:STATE &lt;non-zero NR1&gt;</td>
<td>ON</td>
</tr>
<tr>
<td>:AUTOset &lt; EXECute &gt;</td>
<td></td>
</tr>
<tr>
<td>CALibrate:INTERNal</td>
<td></td>
</tr>
<tr>
<td>CALibrate:INTERNal:STARt</td>
<td></td>
</tr>
<tr>
<td>CALibrate:FACtory STARt</td>
<td></td>
</tr>
<tr>
<td>CALibrate:FACtory CONTinue</td>
<td></td>
</tr>
<tr>
<td>CALibrate:FACtory PREVious</td>
<td></td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:AUTOZero EXECute</td>
<td></td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:DEGAUss EXECute</td>
<td></td>
</tr>
<tr>
<td>DIAg:STATE EXECute</td>
<td></td>
</tr>
<tr>
<td>FACTory</td>
<td></td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt; :RESUIts</td>
<td>When used in single sequence acquisition mode or during waveform recall.</td>
</tr>
<tr>
<td>RECALL:SETUp (&lt;file as quoted string&gt;</td>
<td>FACtory</td>
</tr>
<tr>
<td>RECALL:WAVEform &lt;.ISF or .CSV file &gt;,&lt; REF&lt;x&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td>:RF:REFLevel AUTO</td>
<td></td>
</tr>
<tr>
<td>*RST</td>
<td></td>
</tr>
<tr>
<td>SAVe:IMAGe &lt;file as quoted string&gt;</td>
<td></td>
</tr>
<tr>
<td>SAVe:SETUp &lt;file as quoted string&gt;</td>
<td></td>
</tr>
<tr>
<td>SAVe:WAVEform &lt; source wfm &gt;, (&lt; REF&lt;x&gt; &gt;</td>
<td>&lt; file &gt;)</td>
</tr>
<tr>
<td>TEKSecure</td>
<td></td>
</tr>
<tr>
<td>TRIGger:A SETLevel</td>
<td></td>
</tr>
</tbody>
</table>
For example, a typical application might involve acquiring a single-sequence waveform and then taking a measurement on the acquired waveform. You could use the following command sequence to do this:

```plaintext
/** Set up conditional acquisition **/
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/** Acquire waveform data **/
ACQUIRE:STATE ON
/** Set up the measurement parameters **/
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/** Take amplitude measurement **/
MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MEAN?
```

The acquisition of the waveform requires extended processing time. It may not finish before the oscilloscope takes an amplitude measurement (see the following figure). This can result in an incorrect amplitude value.

**Figure 3-7: Command Processing Without Using Synchronization**

To ensure the oscilloscope completes waveform acquisition before taking the measurement on the acquired data, you can synchronize the program using *WAI, BUSY, *OPC, and *OPC?.

**Figure 3-8: Processing Sequence With Synchronization**

**Example of Using the *OPC Command**

If the corresponding status registers are enabled, the *OPC command sets the OPC bit in the Standard Event Status Register (SESR) when an operation is complete. You achieve synchronization by using this command with either a serial poll or service request handler.
**Serial Poll Method:** Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and *ESE commands.

When the operation is complete, the OPC bit in the Standard Event Status Register (SESR) will be enabled and the Event Status Bit (ESB) in the Status Byte Register will be enabled.

The same command sequence using the *OPC command for synchronization with serial polling looks like this:

```plaintext
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Enable the status registers */
DESE 1
*ESE 1
*SRE 0
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters on the channel we're about to sequence */
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/* Wait until the acquisition is complete before taking the measurement. */
*OPC
While serial poll = 0, keep looping
/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESULT:CURRENTACQ:MEAN?
```

This technique requires less bus traffic than did looping on BUSY.

**Service Request Method:** Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and *ESE commands.

You can also enable service requests by setting the ESB bit in the Service Request Enable Register (SRER) using the *SRE command. When the operation is complete, the oscilloscope will generate a Service Request.

The same command sequence using the *OPC command for synchronization looks like this:

```plaintext
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
```
/* Enable the status registers */
DESE 1
*ESE 1
*SRE 32

/* Set up the measurement parameters on the channel we're about to sequence */
:MEASurement:MEAS1:TYPE AMPLITUDE
:MEASurement:MEAS1:SOURCE CH1
/* Acquire waveform data */
ACQUIRE:STATE ON

/* Wait until the acquisition is complete before taking the measurement*/
*OPC

The program can now do different tasks such as talk to other devices. The SRQ, when it comes, interrupts those tasks and returns control to this task.

/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESUlts:CURREntacq:MEAN?

---

**Example of Using the *OPC? Query**

The *OPC? query places a 1 in the Output Queue once an operation that generates an OPC message is complete. The *OPC? query does not return until all pending OPC operations have completed. Therefore, your time-out must be set to a time at least as long as the longest expected time for the operations to complete.

The same command sequence using the *OPC? query for synchronization looks like this:

/* Set up single sequence acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Set up the measurement parameters on the channel we're about to sequence */
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Wait until the acquisition is complete before taking the measurement*/
*OPC?

Wait for read from Output Queue.
/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESUlts:CURREntacq:MEAN?
This is the simplest approach. It requires no status handling or loops. However, you must set the controller time-out for longer than the acquisition operation.

Example of Using the *WAI Command

The *WAI command forces completion of previous commands that generate an OPC message. No commands after the *WAI are processed before the OPC message(s) are generated.

The same command sequence using the *WAI command for synchronization looks like this:

```plaintext
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Set up the measurement parameters on the channel we're about to sequence */
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Wait until the acquisition is complete before taking the measurement*/
*/
*WAI
/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MEAN?
```

The controller can continue to write commands to the input buffer of the oscilloscope, but the commands will not be processed by the oscilloscope until all in-process OPC operations are complete. If the input buffer becomes full, the controller will be unable to write commands to the buffer. This can cause a time-out.

Example of Using the BUSY Query

The BUSY? query allows you to find out whether the oscilloscope is busy processing a command that has an extended processing time such as single-sequence acquisition.

The same command sequence, using the BUSY? query for synchronization, looks like this:

```plaintext
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
```
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* Wait until the acquisition is complete before taking
the measurement */
while BUSY? keep looping
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE?

This sequence lets you create your own wait loop rather than using the *WAI
command. The BUSY? query helps you avoid time-outs caused by writing too
many commands to the input buffer. The controller is still tied up though, and
the repeated BUSY? query will result in bus traffic.

Reference waveforms

Measurements on references also support OPC when used in conjunction with a
:RECALL:WAVEFORM command.

/* Add a reference slot on which to measure */
:REF:ADDNEW "REF1"
/* Set up the measurement parameters on the reference */
:MEASUREMENT:MEAS1:TYPE AMPLITUDE
:MEASUREMENT:MEAS1:SOURCE REF1
/* Load the new waveform file */
:RECALL:WAVEFORM "E:\waveform.wfm",REF1
*OPC?
/* Wait for read from Output Queue. */
/* Take amplitude measurement */
:MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MEAN?

Messages

The information contained in the topics above covers all the programming
interface messages the oscilloscope generates in response to commands and
queries.

For most messages, a secondary message from the oscilloscope gives detail about
the cause of the error or the meaning of the message. This message is part of the
message string and is separated from the main message by a semicolon.

Each message is the result of an event. Each type of event sets a specific bit in the
SESR and is controlled by the equivalent bit in the DESER. Thus, each message
is associated with a specific SESR bit. In the message tables, the associated SESR bit is specified in the table title, with exceptions noted with the error message text.

**No Event**

The following table shows the messages when the system has no events or status to report. These have no associated SESR bit.

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No events to report; queue empty</td>
</tr>
<tr>
<td>1</td>
<td>No events to report; new events pending *ESR?</td>
</tr>
</tbody>
</table>

**Command Error**

The following table shows the command error messages generated by improper syntax. Check that the command is properly formed and that it follows the rules in the section on command Syntax.

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Command error</td>
</tr>
<tr>
<td>101</td>
<td>Invalid character</td>
</tr>
<tr>
<td>102</td>
<td>Syntax error</td>
</tr>
<tr>
<td>103</td>
<td>Invalid separator</td>
</tr>
<tr>
<td>104</td>
<td>Data type error</td>
</tr>
<tr>
<td>105</td>
<td>GET not allowed</td>
</tr>
<tr>
<td>108</td>
<td>Parameter not allowed</td>
</tr>
<tr>
<td>109</td>
<td>Missing parameter</td>
</tr>
<tr>
<td>110</td>
<td>Command header error</td>
</tr>
<tr>
<td>112</td>
<td>Program mnemonic too long</td>
</tr>
<tr>
<td>113</td>
<td>Undefined header</td>
</tr>
<tr>
<td>120</td>
<td>Numeric data error</td>
</tr>
<tr>
<td>121</td>
<td>Invalid character in numeric</td>
</tr>
<tr>
<td>123</td>
<td>Exponent too large</td>
</tr>
<tr>
<td>124</td>
<td>Too many digits</td>
</tr>
<tr>
<td>130</td>
<td>Suffix error</td>
</tr>
<tr>
<td>131</td>
<td>Invalid suffix</td>
</tr>
<tr>
<td>134</td>
<td>Suffix too long</td>
</tr>
<tr>
<td>140</td>
<td>Character data error</td>
</tr>
<tr>
<td>141</td>
<td>Invalid character data</td>
</tr>
<tr>
<td>144</td>
<td>Character data too long</td>
</tr>
<tr>
<td>150</td>
<td>String data error</td>
</tr>
</tbody>
</table>
### Table 3-5: Command Error Messages (CME Bit 5) (cont.)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>151</td>
<td>Invalid string data</td>
</tr>
<tr>
<td>152</td>
<td>String data too long</td>
</tr>
<tr>
<td>160</td>
<td>Block data error</td>
</tr>
<tr>
<td>161</td>
<td>Invalid block data</td>
</tr>
<tr>
<td>170</td>
<td>Command expression error</td>
</tr>
<tr>
<td>171</td>
<td>Invalid expression</td>
</tr>
</tbody>
</table>

### Execution Error

The following table lists the execution errors that are detected during execution of a command.

### Table 3-6: Execution Error Messages (EXE Bit 4)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Execution error</td>
</tr>
<tr>
<td>221</td>
<td>Settings conflict</td>
</tr>
<tr>
<td>222</td>
<td>Data out of range</td>
</tr>
<tr>
<td>224</td>
<td>Illegal parameter value</td>
</tr>
<tr>
<td>241</td>
<td>Hardware missing</td>
</tr>
<tr>
<td>250</td>
<td>Mass storage error</td>
</tr>
<tr>
<td>251</td>
<td>Missing mass storage</td>
</tr>
<tr>
<td>252</td>
<td>Missing media</td>
</tr>
<tr>
<td>253</td>
<td>Corrupt media</td>
</tr>
<tr>
<td>254</td>
<td>Media full</td>
</tr>
<tr>
<td>255</td>
<td>Directory full</td>
</tr>
<tr>
<td>256</td>
<td>File name not found</td>
</tr>
<tr>
<td>257</td>
<td>File name error</td>
</tr>
<tr>
<td>258</td>
<td>Media protected</td>
</tr>
<tr>
<td>259</td>
<td>File name too long</td>
</tr>
<tr>
<td>280</td>
<td>Program error</td>
</tr>
<tr>
<td>282</td>
<td>Insufficient network printer information</td>
</tr>
<tr>
<td>283</td>
<td>Network printer not responding</td>
</tr>
<tr>
<td>284</td>
<td>Network printer server not responding</td>
</tr>
<tr>
<td>286</td>
<td>Program runtime error</td>
</tr>
<tr>
<td>287</td>
<td></td>
</tr>
<tr>
<td>2200</td>
<td>Measurement error, Measurement system error</td>
</tr>
<tr>
<td>2201</td>
<td>Measurement error, Zero period</td>
</tr>
<tr>
<td>2202</td>
<td>Measurement error, No period, second waveform</td>
</tr>
</tbody>
</table>
Table 3-6: Execution Error Messages (EXE Bit 4) (cont.)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>2203</td>
<td>Measurement error, No period, second waveform</td>
</tr>
<tr>
<td>2204</td>
<td>Measurement error, Low amplitude, second waveform</td>
</tr>
<tr>
<td>2205</td>
<td>Measurement error, Low amplitude, second waveform</td>
</tr>
<tr>
<td>2206</td>
<td>Measurement error, Invalid gate</td>
</tr>
<tr>
<td>2207</td>
<td>Measurement error, Measurement overflow</td>
</tr>
<tr>
<td>2208</td>
<td>Measurement error, No backwards Mid Ref crossing</td>
</tr>
<tr>
<td>2209</td>
<td>Measurement error, No second Mid Ref crossing</td>
</tr>
<tr>
<td>2210</td>
<td>Measurement error, No Mid Ref crossing, second waveform</td>
</tr>
<tr>
<td>2211</td>
<td>Measurement error, No backwards Mid Ref crossing</td>
</tr>
<tr>
<td>2212</td>
<td>Measurement error, No negative crossing</td>
</tr>
<tr>
<td>2213</td>
<td>Measurement error, No positive crossing</td>
</tr>
<tr>
<td>2214</td>
<td>Measurement error, No crossing, target waveform</td>
</tr>
<tr>
<td>2215</td>
<td>Measurement error, No crossing, second waveform</td>
</tr>
<tr>
<td>2216</td>
<td>Measurement error, No crossing, target waveform</td>
</tr>
<tr>
<td>2217</td>
<td>Measurement error, Constant waveform</td>
</tr>
<tr>
<td>2219</td>
<td>Measurement error, No valid edge - No arm sample</td>
</tr>
<tr>
<td>2220</td>
<td>Measurement error, No valid edge - No arm cross</td>
</tr>
<tr>
<td>2221</td>
<td>Measurement error, No valid edge - No trigger cross</td>
</tr>
<tr>
<td>2222</td>
<td>Measurement error, No valid edge - No second cross</td>
</tr>
<tr>
<td>2223</td>
<td>Measurement error, Waveform mismatch</td>
</tr>
<tr>
<td>2224</td>
<td>Measurement error, WAIT calculating</td>
</tr>
<tr>
<td>2225</td>
<td>Measurement error, No waveform to measure</td>
</tr>
<tr>
<td>2226</td>
<td>Measurement error, Null Waveform</td>
</tr>
<tr>
<td>2227</td>
<td>Measurement error, Positive and Negative Clipping</td>
</tr>
<tr>
<td>2228</td>
<td>Measurement error, Positive Clipping</td>
</tr>
<tr>
<td>2229</td>
<td>Measurement error, Negative Clipping</td>
</tr>
<tr>
<td>2230</td>
<td>Measurement error, High Ref &lt; Low Ref</td>
</tr>
<tr>
<td>2231</td>
<td>Measurement error, No statistics available</td>
</tr>
<tr>
<td>2233</td>
<td>Requested waveform is temporarily unavailable</td>
</tr>
<tr>
<td>2235</td>
<td>Math error, invalid math description</td>
</tr>
<tr>
<td>2240</td>
<td>Invalid password</td>
</tr>
<tr>
<td>2241</td>
<td>Waveform requested is invalid</td>
</tr>
<tr>
<td>2244</td>
<td>Source waveform is not active</td>
</tr>
<tr>
<td>2245</td>
<td>Saveref error, selected channel is turned off</td>
</tr>
<tr>
<td>2250</td>
<td>Reference error, the reference waveform file is invalid</td>
</tr>
<tr>
<td>2253</td>
<td>Reference error, too many points received</td>
</tr>
<tr>
<td>2254</td>
<td>Reference error, too few points received</td>
</tr>
</tbody>
</table>
### Table 3-6: Execution Error Messages (EXE Bit 4) (cont.)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>2259</td>
<td>File too big</td>
</tr>
<tr>
<td>2270</td>
<td>Alias error</td>
</tr>
<tr>
<td>2271</td>
<td>Alias syntax error</td>
</tr>
<tr>
<td>2273</td>
<td>Illegal alias label</td>
</tr>
<tr>
<td>2276</td>
<td>Alias expansion error</td>
</tr>
<tr>
<td>2277</td>
<td>Alias redefinition not allowed</td>
</tr>
<tr>
<td>2278</td>
<td>Alias header not found</td>
</tr>
<tr>
<td>2285</td>
<td>TekSecure(R) Pass</td>
</tr>
<tr>
<td>2286</td>
<td>TekSecure(R) Fail</td>
</tr>
<tr>
<td>2500</td>
<td>Setup error, file does not look like a setup file</td>
</tr>
<tr>
<td>2501</td>
<td>Setup warning, could not recall all values from external setup</td>
</tr>
<tr>
<td>2620</td>
<td>Mask error, too few points received</td>
</tr>
<tr>
<td>2760</td>
<td>Mark limit reached</td>
</tr>
<tr>
<td>2761</td>
<td>No mark present</td>
</tr>
<tr>
<td>2762</td>
<td>Search copy failed</td>
</tr>
</tbody>
</table>

### Device Error

The following table lists the device errors that can occur during oscilloscope operation. These errors may indicate that the oscilloscope needs repair.

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>310</td>
<td>System error</td>
</tr>
<tr>
<td>311</td>
<td>Memory error</td>
</tr>
<tr>
<td>312</td>
<td>PUD memory lost</td>
</tr>
<tr>
<td>314</td>
<td>Save/recall memory lost</td>
</tr>
</tbody>
</table>

### System Event

The following table lists the system event messages. These messages are generated whenever certain system conditions occur.

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Query event</td>
</tr>
<tr>
<td>401</td>
<td>Power on (PON bit 7 set)</td>
</tr>
<tr>
<td>402</td>
<td>Operation complete (OPC bit 0 set)</td>
</tr>
<tr>
<td>403</td>
<td>User request (URQ bit 6 set)</td>
</tr>
<tr>
<td>404</td>
<td>Power fail (DDE bit 3 set)</td>
</tr>
</tbody>
</table>
Table 3-8: System Event Messages (cont.)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>405</td>
<td>Request control</td>
</tr>
<tr>
<td>410</td>
<td>Query INTERRUPTED (QYE bit 2 set)</td>
</tr>
<tr>
<td>420</td>
<td>Query UNTERMINATED (QYE bit 2 set)</td>
</tr>
<tr>
<td>430</td>
<td>Query DEADLOCKED (QYE bit 2 set)</td>
</tr>
<tr>
<td>440</td>
<td>Query UNTERMINATED after indefinite response (QYE bit 2 set)</td>
</tr>
<tr>
<td>468</td>
<td>Knob/Keypad value changed</td>
</tr>
<tr>
<td>472</td>
<td>Application variable changed</td>
</tr>
</tbody>
</table>

Execution Warning

The following table lists warning messages that do not interrupt the flow of command execution. They also notify you of possible unexpected results.

Table 3-9: Execution Warning Messages (EXE Bit 4)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>528</td>
<td>Parameter out of range</td>
</tr>
<tr>
<td>532</td>
<td>Curve data too long, Curve truncated</td>
</tr>
<tr>
<td>533</td>
<td>Curve error, Preamble values are inconsistent</td>
</tr>
<tr>
<td>540</td>
<td>Measurement warning, Uncertain edge</td>
</tr>
<tr>
<td>541</td>
<td>Measurement warning, Low signal amplitude</td>
</tr>
<tr>
<td>542</td>
<td>Measurement warning, Unstable histogram</td>
</tr>
<tr>
<td>543</td>
<td>Measurement warning, Low resolution</td>
</tr>
<tr>
<td>544</td>
<td>Measurement warning, Uncertain edge</td>
</tr>
<tr>
<td>545</td>
<td>Measurement warning, Invalid in minmax</td>
</tr>
<tr>
<td>546</td>
<td>Measurement warning, Need 3 edges</td>
</tr>
<tr>
<td>547</td>
<td>Measurement warning, Clipping positive/negative</td>
</tr>
<tr>
<td>548</td>
<td>Measurement warning, Clipping positive</td>
</tr>
<tr>
<td>549</td>
<td>Measurement warning, Clipping negative</td>
</tr>
</tbody>
</table>

Table 3-10: Execution Warning Messages (EXE Bit 4)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>540</td>
<td>Measurement warning</td>
</tr>
<tr>
<td>541</td>
<td>Measurement warning, Low signal amplitude</td>
</tr>
<tr>
<td>542</td>
<td>Measurement warning, Unstable histogram</td>
</tr>
<tr>
<td>543</td>
<td>Measurement warning, Low resolution</td>
</tr>
<tr>
<td>544</td>
<td>Measurement warning, Uncertain edge</td>
</tr>
<tr>
<td>545</td>
<td>Measurement warning, Invalid min max</td>
</tr>
<tr>
<td>546</td>
<td>Measurement warning, Need 3 edges</td>
</tr>
</tbody>
</table>
### Table 3-10: Execution Warning Messages (EXE Bit 4) (cont.)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>547</td>
<td>Measurement warning, Clipping positive/negative</td>
</tr>
<tr>
<td>548</td>
<td>Measurement warning, Clipping positive</td>
</tr>
<tr>
<td>549</td>
<td>Measurement warning, Clipping negative</td>
</tr>
</tbody>
</table>

### Internal Warning

The following table shows internal errors that indicate an internal fault in the oscilloscope.

#### Table 3-11: Internal Warning Messages

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>630</td>
<td>Internal warning, 50Ω overload</td>
</tr>
</tbody>
</table>
## Appendix A: Character Set

<table>
<thead>
<tr>
<th>B7 B6 B5 B4 B3 B2 B1</th>
<th>0 0 0 0</th>
<th>0 0 0 1</th>
<th>0 0 1 0</th>
<th>0 0 1 1</th>
<th>0 1 0 0</th>
<th>0 1 0 1</th>
<th>0 1 1 0</th>
<th>0 1 1 1</th>
<th>1 0 0 0</th>
<th>1 0 0 1</th>
<th>1 0 1 0</th>
<th>1 0 1 1</th>
<th>1 1 0 0</th>
<th>1 1 0 1</th>
<th>1 1 1 0</th>
<th>1 1 1 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CONTROL</strong></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td><strong>NUMBERS SYMBOLS</strong></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td><strong>UPPER CASE</strong></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
<tr>
<td><strong>LOWER CASE</strong></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
</tbody>
</table>

### Key
- **Octal**: 5
- **Hex**: 5
- **GPIB code (with ATN asserted)**
- **ASCII character**: decimal

---

**Notes**

- MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer

---

**Tektronix**

REF: ANSI STD X3.4-1977
IEEE STD 488.1-1987
ISO STD 646-2973
Appendix B: Reserved Words

This is a list of reserved words for your instrument. Capital letters identify the required minimum spelling.

<table>
<thead>
<tr>
<th>Reserved Word</th>
<th>Meaning</th>
<th>Reserved Word</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>*CAL</td>
<td>ADVanced</td>
<td>AUTOINCrement</td>
<td>BATHTUB</td>
</tr>
<tr>
<td>*CLS</td>
<td>AFG</td>
<td>AUTORange</td>
<td>BATHTub</td>
</tr>
<tr>
<td>*DDT</td>
<td>AHPOsition</td>
<td>AUTOSAVEPITIMEOUT</td>
<td>BCR</td>
</tr>
<tr>
<td>*ESE</td>
<td>ALlAs</td>
<td>AUTOSAVEUETIMEOUT</td>
<td>BDIFFBP</td>
</tr>
<tr>
<td>*ESR</td>
<td>ALL</td>
<td>AUTOscale</td>
<td>BEGIN</td>
</tr>
<tr>
<td>*IDN</td>
<td>ALLAqcS</td>
<td>AUTOset</td>
<td>BELL</td>
</tr>
<tr>
<td>*LRN</td>
<td>ALLBits</td>
<td>AUTOZero</td>
<td>BER</td>
</tr>
<tr>
<td>*OPC</td>
<td>ALEV</td>
<td>AUTOMATIC</td>
<td>BHPOsition</td>
</tr>
<tr>
<td>*OPT</td>
<td>ALLQstring</td>
<td>AUTOscale</td>
<td>BIAS</td>
</tr>
<tr>
<td>*PSC</td>
<td>ALLTHresholds</td>
<td>AUTOset</td>
<td>BIN</td>
</tr>
<tr>
<td>*PUD</td>
<td>ALTERNATEA</td>
<td>AUXLevel</td>
<td>BINARY</td>
</tr>
<tr>
<td>*RST</td>
<td>ALTERNATEB</td>
<td>AUXiliary</td>
<td>BINARY</td>
</tr>
<tr>
<td>*SRE</td>
<td>ALWAYS</td>
<td>AUXout</td>
<td>BIT</td>
</tr>
<tr>
<td>*STB</td>
<td>AM14</td>
<td>AVAILABLE</td>
<td>BITAMPLITUDE</td>
</tr>
<tr>
<td>*TRG</td>
<td>AMPLitude</td>
<td>AVERAGE</td>
<td>BITcfgmode</td>
</tr>
<tr>
<td>*TST</td>
<td>AMPLITUDE</td>
<td>AVERAGE</td>
<td>BITDelay</td>
</tr>
<tr>
<td>*WAI</td>
<td>ANALOG</td>
<td>AVG</td>
<td>BITEnd</td>
</tr>
<tr>
<td>0</td>
<td>ANALog</td>
<td>BITHIGH</td>
<td>BITOrder</td>
</tr>
<tr>
<td>1</td>
<td>ANALYZemode</td>
<td>BITYlow</td>
<td>BJT</td>
</tr>
<tr>
<td>7</td>
<td>AND</td>
<td>BIT_Nr</td>
<td>BLACKMANHarris</td>
</tr>
<tr>
<td>8</td>
<td>ANNOTate</td>
<td>B0</td>
<td>BITRate</td>
</tr>
<tr>
<td>9</td>
<td>ANY</td>
<td>B1</td>
<td>BITSTUFFing</td>
</tr>
<tr>
<td>A</td>
<td>ANYERRor</td>
<td>B10</td>
<td>BITStart</td>
</tr>
<tr>
<td>ABOrt</td>
<td>AoffSet</td>
<td>B11</td>
<td>BITType</td>
</tr>
<tr>
<td>ABSolute</td>
<td>APPPID</td>
<td>B12</td>
<td>BIT_Nr</td>
</tr>
<tr>
<td>AC</td>
<td>APPly</td>
<td>B13</td>
<td>BJT</td>
</tr>
<tr>
<td>ACCOMMONMODE</td>
<td>APPPW</td>
<td>B14</td>
<td>BLANKingtime</td>
</tr>
<tr>
<td>ACCDRMS</td>
<td>ARBITRARY</td>
<td>B15</td>
<td>block_data</td>
</tr>
<tr>
<td>ACK</td>
<td>AREA</td>
<td>B16</td>
<td>BM</td>
</tr>
<tr>
<td>ACKnum</td>
<td>AREAOfcrosssection</td>
<td>B2</td>
<td>BMP</td>
</tr>
<tr>
<td>ACQ</td>
<td>ARINC429A</td>
<td>B3</td>
<td>BN_Fmt</td>
</tr>
<tr>
<td>ACQdurATION</td>
<td>ASCII</td>
<td>B4</td>
<td>BOFFset</td>
</tr>
<tr>
<td>ACQQUIRE</td>
<td>ASCL</td>
<td>B5</td>
<td>BOLD</td>
</tr>
<tr>
<td>ACQquisition</td>
<td>ASC_Fmt</td>
<td>B6</td>
<td>boolean</td>
</tr>
<tr>
<td>ACRMS</td>
<td>ASCii</td>
<td>B7</td>
<td>BOTH</td>
</tr>
<tr>
<td>ACTive</td>
<td>ASCurve</td>
<td>B8</td>
<td>BPOSition</td>
</tr>
<tr>
<td>ACTivesourcecount</td>
<td>ASIC</td>
<td>B9</td>
<td>BRSBit</td>
</tr>
<tr>
<td>ADD</td>
<td>ASIS</td>
<td>BACKLight</td>
<td>BUS</td>
</tr>
<tr>
<td>ADDMEAS</td>
<td>ASOURCE</td>
<td>BACKward</td>
<td>BUSAndWAVEFORM</td>
</tr>
<tr>
<td>ADDNew</td>
<td>ATRIGger</td>
<td>BANDwidth</td>
<td>BULK</td>
</tr>
<tr>
<td>ADDR10</td>
<td>AUDIO</td>
<td>BANDwidth</td>
<td>BUNIts</td>
</tr>
<tr>
<td>ADDR7</td>
<td>AUNITS</td>
<td>BASE</td>
<td>BURSTEDGTYPE</td>
</tr>
<tr>
<td>ADDR1ANDDATA</td>
<td>AUTHenticate</td>
<td>BASETop</td>
<td>BURSTWIDTH</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>AUTO</td>
<td>BASEline</td>
<td>BUS</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>AUTODim</td>
<td>BASIC</td>
<td>BUSANDWAVEFORM</td>
</tr>
<tr>
<td>Reserved Word</td>
<td>Reserved Word</td>
<td>Reserved Word</td>
<td>Reserved Word</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
<td>---------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>CLOCKSOURCE</td>
<td>CYCLEMAX</td>
<td>DDRTRERRN</td>
<td></td>
</tr>
<tr>
<td>CLOCK</td>
<td>CYCLEMin</td>
<td>DDRTJITCC</td>
<td></td>
</tr>
<tr>
<td>BUSY</td>
<td>CYCLEMode</td>
<td>DDRTJITDUTY</td>
<td></td>
</tr>
<tr>
<td>BUSx</td>
<td>CYCLEPKP</td>
<td>DDRTJITEW</td>
<td></td>
</tr>
<tr>
<td>BVPOSITION</td>
<td>CYCLETEop</td>
<td>DDRTPST</td>
<td></td>
</tr>
<tr>
<td>BXPOSITION</td>
<td>CYCLEcount</td>
<td>DDRTPRE</td>
<td></td>
</tr>
<tr>
<td>BY</td>
<td>CYCE历代</td>
<td>DDRTWPRE</td>
<td></td>
</tr>
<tr>
<td>BYPOSITION</td>
<td>Chx_Dx</td>
<td>DDRVIXAC</td>
<td></td>
</tr>
<tr>
<td>BYT_Nr</td>
<td>Current</td>
<td>DDRWIRE</td>
<td></td>
</tr>
<tr>
<td>BYT_Or</td>
<td>DAMping</td>
<td>DDT</td>
<td></td>
</tr>
<tr>
<td>Block</td>
<td>DASHED</td>
<td>DECIMAL</td>
<td></td>
</tr>
<tr>
<td>Bpeak</td>
<td>DASHSOLID</td>
<td>DEFaultsetup</td>
<td></td>
</tr>
<tr>
<td>Br</td>
<td>DATA</td>
<td>DEFINE</td>
<td></td>
</tr>
<tr>
<td>Bus</td>
<td>DATA0</td>
<td>DEGAUSS</td>
<td></td>
</tr>
<tr>
<td>Bx</td>
<td>DATA1</td>
<td>DEGREes</td>
<td></td>
</tr>
<tr>
<td>CALibrate</td>
<td>DATA2</td>
<td>DELAY</td>
<td></td>
</tr>
<tr>
<td>CAN</td>
<td>DATAbits</td>
<td>DELETE</td>
<td></td>
</tr>
<tr>
<td>CAN2X</td>
<td>DATAFORMAT</td>
<td>DELIMITER</td>
<td></td>
</tr>
<tr>
<td>CANH</td>
<td>DATAMEASUREMENT</td>
<td>DELTatime</td>
<td></td>
</tr>
<tr>
<td>CANL</td>
<td>DATAPacket</td>
<td>DELAY</td>
<td></td>
</tr>
<tr>
<td>CAPACITANCE</td>
<td>DATAPath</td>
<td>DELETE</td>
<td></td>
</tr>
<tr>
<td>CARDiac</td>
<td>DATARESULTS</td>
<td>DELEleteqstring</td>
<td></td>
</tr>
<tr>
<td>CATalog</td>
<td>CDCAPACITANCE</td>
<td>DESE</td>
<td></td>
</tr>
<tr>
<td>CCRESULTS</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCycles</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CENTERFREQUENCY</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CGS</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CH</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CH1</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHAN1A</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHAN2B</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHANwidth</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHX</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channel</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Checksum</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CH</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CH_DALL</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CH1_D</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHX</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHX_DALL</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHX_DX</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLASSA</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLASSB</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLASSC</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLASSD</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLAss</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLEAR</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLOCK</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLOCKBITSPERCHANNEL</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLOCKFrequency</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLOCKMultiplier</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLOCKRecovery</td>
<td>CDCAPACITANCE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
<table>
<thead>
<tr>
<th>Reserved Word</th>
<th>Appendix B: Reserved Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>DJDD</td>
<td>DJDIRAC</td>
</tr>
<tr>
<td>DJDIRAC</td>
<td>EOW</td>
</tr>
<tr>
<td>MINus</td>
<td>EQUAL</td>
</tr>
<tr>
<td>DNS</td>
<td>Equal</td>
</tr>
<tr>
<td>DOFFSet</td>
<td>Equal$\neq$Equal</td>
</tr>
<tr>
<td>DOMAINname</td>
<td>ERISe</td>
</tr>
<tr>
<td>DONTInclude</td>
<td>ERROR</td>
</tr>
<tr>
<td>DOTsonly</td>
<td>ERRTYPE</td>
</tr>
<tr>
<td>DOUBLE</td>
<td>ERRType</td>
</tr>
<tr>
<td>DPLUS</td>
<td>ERRor</td>
</tr>
<tr>
<td>DRIVE</td>
<td>FAST</td>
</tr>
<tr>
<td>Duty</td>
<td>FASTAcq</td>
</tr>
<tr>
<td>DVDT</td>
<td>ET</td>
</tr>
<tr>
<td>DVM</td>
<td>ETHERNET</td>
</tr>
<tr>
<td>DYNAMIC</td>
<td>ETHERnet</td>
</tr>
<tr>
<td>DYNRange</td>
<td>EVEN</td>
</tr>
<tr>
<td>DeltaB</td>
<td>EVENT</td>
</tr>
<tr>
<td>DeltaH</td>
<td>EVENTS</td>
</tr>
<tr>
<td>D</td>
<td>EVENTtable</td>
</tr>
<tr>
<td>EACHCLOCKCYCLE</td>
<td>EVERY</td>
</tr>
<tr>
<td>ECL</td>
<td>EVMsg</td>
</tr>
<tr>
<td>EDECay</td>
<td>F2</td>
</tr>
<tr>
<td>EDGE</td>
<td>F3MAG</td>
</tr>
<tr>
<td>EDGEIncre</td>
<td>FC1063</td>
</tr>
<tr>
<td>EDGES</td>
<td>FC133</td>
</tr>
<tr>
<td>EDGESOURCE</td>
<td>FC2125</td>
</tr>
<tr>
<td>EDGEType</td>
<td>FC266</td>
</tr>
<tr>
<td>EDGE</td>
<td>FC4250</td>
</tr>
<tr>
<td>EDGE</td>
<td>FC531</td>
</tr>
<tr>
<td>EFFICIENCY</td>
<td>FC8500</td>
</tr>
<tr>
<td>EITHER</td>
<td>FCSError</td>
</tr>
<tr>
<td>EIther</td>
<td>FROM</td>
</tr>
<tr>
<td>ENABLE</td>
<td>FROMLevel</td>
</tr>
<tr>
<td>ENable</td>
<td>FROMSearch</td>
</tr>
<tr>
<td>ENCdg</td>
<td>FROMMedge</td>
</tr>
<tr>
<td>END</td>
<td>FROMReference</td>
</tr>
<tr>
<td>ENDPoint</td>
<td>FROMReference</td>
</tr>
<tr>
<td>ENET</td>
<td>FROMReference</td>
</tr>
<tr>
<td>ENET100</td>
<td>FROMReference</td>
</tr>
<tr>
<td>ENET1000</td>
<td>FULL</td>
</tr>
<tr>
<td>ENGINEering</td>
<td>FULL</td>
</tr>
<tr>
<td>ENHANCED4</td>
<td>FULL</td>
</tr>
<tr>
<td>ENHANCED8</td>
<td>FULL</td>
</tr>
<tr>
<td>ENHanced</td>
<td>FULL</td>
</tr>
<tr>
<td>ENTERSWindow</td>
<td>FULL</td>
</tr>
<tr>
<td>Env</td>
<td>FULLColor</td>
</tr>
<tr>
<td>Envelope</td>
<td>FULLSPEED</td>
</tr>
<tr>
<td>EOF</td>
<td>FULLScreen</td>
</tr>
<tr>
<td>EOFTYPE</td>
<td>FOCUS</td>
</tr>
<tr>
<td>EOP</td>
<td>FOCUS</td>
</tr>
<tr>
<td>FAIL</td>
<td>FOLDER</td>
</tr>
<tr>
<td>FALL</td>
<td>FONT</td>
</tr>
<tr>
<td>FALLHigh</td>
<td>FORCEDRange</td>
</tr>
<tr>
<td>FALLING</td>
<td>FORCe</td>
</tr>
<tr>
<td>FALLLow</td>
<td>FORMERRor</td>
</tr>
<tr>
<td>FALLMid</td>
<td>FORMAT</td>
</tr>
<tr>
<td>FALSE</td>
<td>FORWARD</td>
</tr>
<tr>
<td>FALL</td>
<td>FOUR</td>
</tr>
<tr>
<td>FALLing</td>
<td>FOURHundred</td>
</tr>
<tr>
<td>FAST</td>
<td>FOURTEENTen</td>
</tr>
<tr>
<td>FASTAcq</td>
<td>FP</td>
</tr>
<tr>
<td>FASTERthan</td>
<td>FPanel</td>
</tr>
<tr>
<td>FASTER</td>
<td>FBinary</td>
</tr>
<tr>
<td>FASTframe</td>
<td>FRAME</td>
</tr>
<tr>
<td>FBD1</td>
<td>FRAMEID</td>
</tr>
<tr>
<td>FBD2</td>
<td>FRAMELENghth</td>
</tr>
<tr>
<td>FBD3</td>
<td>FRAMETYPE</td>
</tr>
<tr>
<td>FRAMING</td>
<td>FRAME</td>
</tr>
<tr>
<td>FRAME</td>
<td>FRAMType</td>
</tr>
<tr>
<td>FC1063</td>
<td>FTP</td>
</tr>
<tr>
<td>FC133</td>
<td>FRMe</td>
</tr>
<tr>
<td>FC2125</td>
<td>FramE</td>
</tr>
<tr>
<td>FC266</td>
<td>FramE</td>
</tr>
<tr>
<td>FC4250</td>
<td>FREference</td>
</tr>
<tr>
<td>FC531</td>
<td>FREQ</td>
</tr>
<tr>
<td>FC8500</td>
<td>FREQUENCY</td>
</tr>
<tr>
<td>FCSError</td>
<td>FREquency</td>
</tr>
<tr>
<td>FROM</td>
<td>FROM</td>
</tr>
<tr>
<td>FROMDIRECT</td>
<td>FROMSearch</td>
</tr>
<tr>
<td>FROMLevel</td>
<td>FROMMedge</td>
</tr>
<tr>
<td>FROMSearch</td>
<td>FROMReference</td>
</tr>
<tr>
<td>FROMMedge</td>
<td>FROMReference</td>
</tr>
<tr>
<td>FIBREchannel</td>
<td>FRam</td>
</tr>
<tr>
<td>FFT</td>
<td>FULL</td>
</tr>
<tr>
<td>FFOne</td>
<td>FULLColor</td>
</tr>
<tr>
<td>FILE</td>
<td>FULLSPEED</td>
</tr>
<tr>
<td>FILEpath</td>
<td>FULLSCREEN</td>
</tr>
<tr>
<td>FILESystem</td>
<td>FULL</td>
</tr>
<tr>
<td>FILEName</td>
<td>FULL</td>
</tr>
<tr>
<td>FILE</td>
<td>FUNCTION</td>
</tr>
<tr>
<td>FILTER</td>
<td>FUNDCurrent</td>
</tr>
<tr>
<td>FILTERs</td>
<td>FW1394BS1600B</td>
</tr>
<tr>
<td>FILEpath</td>
<td>FW1394BS400B</td>
</tr>
<tr>
<td>FIRST</td>
<td>FW1394BS800B</td>
</tr>
<tr>
<td>FIVE</td>
<td>GAP</td>
</tr>
<tr>
<td>FIVEHundred</td>
<td>GATESource</td>
</tr>
<tr>
<td>FIXED</td>
<td>GATEWay</td>
</tr>
<tr>
<td>FLATTOP2</td>
<td>GATing</td>
</tr>
<tr>
<td>FLAG</td>
<td>GAUSSian</td>
</tr>
<tr>
<td>FLEXRAY</td>
<td>GAUSSian</td>
</tr>
<tr>
<td>FLEXray</td>
<td>GLOBAL</td>
</tr>
<tr>
<td>FLYBACK</td>
<td>GLOBALref</td>
</tr>
<tr>
<td>FOCUS</td>
<td>GLOBAL</td>
</tr>
<tr>
<td>FACTory</td>
<td>GLOBAL</td>
</tr>
</tbody>
</table>

MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
<table>
<thead>
<tr>
<th>Reserved Word</th>
<th>Reserved Word</th>
<th>Reserved Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMT</td>
<td>HYSTEResis</td>
<td>INVERTed</td>
</tr>
<tr>
<td>GPKNOB1</td>
<td>HYSteresis</td>
<td>LFRej</td>
</tr>
<tr>
<td>GPKNOB2</td>
<td>Hc</td>
<td>LIC</td>
</tr>
<tr>
<td>GRATicule</td>
<td>Hmax</td>
<td>LICENSE</td>
</tr>
<tr>
<td>GRATicule</td>
<td>I2C</td>
<td>INrange</td>
</tr>
<tr>
<td>GRId</td>
<td>I2S</td>
<td>LIMIT</td>
</tr>
<tr>
<td>GROUPDelay</td>
<td>IBA2500</td>
<td>IO</td>
</tr>
<tr>
<td>HAMMing</td>
<td>ICFATOR</td>
<td>IOUT1SOURCE</td>
</tr>
<tr>
<td>HANDSHAKEPacket</td>
<td>ID</td>
<td>IOUT2SOURCE</td>
</tr>
<tr>
<td>HANDSHAKEType</td>
<td>IDANDDATA</td>
<td>IOUT3SOURCE</td>
</tr>
<tr>
<td>HANNing</td>
<td>IDentifier</td>
<td>IPAddress</td>
</tr>
<tr>
<td>HARMONICS</td>
<td>IDFORMAT</td>
<td>IPHeader</td>
</tr>
<tr>
<td>HAVERSINE</td>
<td>IDLE</td>
<td>IPower</td>
</tr>
<tr>
<td>HBArs</td>
<td>IDle</td>
<td>IPvFOUR</td>
</tr>
<tr>
<td>HDR</td>
<td>IEC</td>
<td>IRMS</td>
</tr>
<tr>
<td>HEADER</td>
<td>ILEVELAbs</td>
<td>LIST</td>
</tr>
<tr>
<td>HEADER</td>
<td>ILEVELPct</td>
<td>LIVe</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>IMAGE</td>
<td>ISOALL</td>
</tr>
<tr>
<td>HEIGHTBER</td>
<td>IMAGinary</td>
<td>LJJ</td>
</tr>
<tr>
<td>HEX</td>
<td>IMPEDance</td>
<td>LOG</td>
</tr>
<tr>
<td>HFRej</td>
<td>IN</td>
<td>LOGIC</td>
</tr>
<tr>
<td>HI</td>
<td>INCLUDEREFs</td>
<td>LOGICPattern</td>
</tr>
<tr>
<td>HID</td>
<td>INCLUDE</td>
<td>LOGICAL</td>
</tr>
<tr>
<td>HIGH</td>
<td>INCLUDE</td>
<td>LOGICQUALification</td>
</tr>
<tr>
<td>HIGHLevel</td>
<td>INDBits</td>
<td>LOGICSource</td>
</tr>
<tr>
<td>HIGHLimit</td>
<td>INDEPENDENT</td>
<td>LS</td>
</tr>
<tr>
<td>HIGHPass</td>
<td>INDUCT</td>
<td>LOW</td>
</tr>
<tr>
<td>HIGHREFVoltage</td>
<td>INDUCTANCE</td>
<td>LOWFREQuency</td>
</tr>
<tr>
<td>HIGHERES</td>
<td>INFInite</td>
<td>LOOP</td>
</tr>
<tr>
<td>HIGHTIME</td>
<td>INFmaximum</td>
<td>LOOPBandwidth</td>
</tr>
<tr>
<td>HIGHZ</td>
<td>INFMINimum</td>
<td>LOREntz</td>
</tr>
<tr>
<td>HIres</td>
<td>INFPersist</td>
<td>LOW</td>
</tr>
<tr>
<td>HISTOGRAM</td>
<td>INIT</td>
<td>LOWERFREQuency</td>
</tr>
<tr>
<td>History</td>
<td>INPUT</td>
<td>LOWER</td>
</tr>
<tr>
<td>HITDMVALUE</td>
<td>INPUTCAP</td>
<td>LOWer</td>
</tr>
<tr>
<td>HIVALue</td>
<td>INPUTMode</td>
<td>LOWerthreshold</td>
</tr>
<tr>
<td>HOLD</td>
<td>INPUTSource</td>
<td>LOWFREVoltage</td>
</tr>
<tr>
<td>HOLDTime</td>
<td>INPUTType</td>
<td>Label</td>
</tr>
<tr>
<td>HOLDoff</td>
<td>INRUSH</td>
<td>LOWSPEED</td>
</tr>
<tr>
<td>HOMEdir</td>
<td>INrush</td>
<td>LOWTHreshold</td>
</tr>
<tr>
<td>HORDER</td>
<td>INSIDEgreater</td>
<td>LATCH</td>
</tr>
<tr>
<td>HORIZontalscale</td>
<td>INSTR</td>
<td>LOWTIME</td>
</tr>
<tr>
<td>HORIZ</td>
<td>INSTALL</td>
<td>MA</td>
</tr>
<tr>
<td>HORIZPOS</td>
<td>INTEGER</td>
<td>MAC</td>
</tr>
<tr>
<td>HORZscale</td>
<td>INTENSITY</td>
<td>MACADDress</td>
</tr>
<tr>
<td>Horizontal</td>
<td>INTERNAL</td>
<td>MACLENGTH</td>
</tr>
<tr>
<td>HSource</td>
<td>INTERPRatio</td>
<td>MAG</td>
</tr>
<tr>
<td>HTTPPort</td>
<td>INTERRUPT</td>
<td>MAGLOSS</td>
</tr>
<tr>
<td>HUB</td>
<td>INTERNAL</td>
<td>MAGNETICLOSS</td>
</tr>
<tr>
<td>HUNDREDBASETX</td>
<td>INTERp</td>
<td>MAGNitude</td>
</tr>
<tr>
<td>Hundred</td>
<td>INVERTNIBble</td>
<td>MAGPROPERTY</td>
</tr>
<tr>
<td>Reserved Word</td>
<td>Meaning</td>
<td>Reserved Word</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>MainWindow</td>
<td></td>
<td>Model</td>
</tr>
<tr>
<td>Main</td>
<td></td>
<td>MOREQua</td>
</tr>
<tr>
<td>Manual</td>
<td></td>
<td>MOREQual</td>
</tr>
<tr>
<td>Manual</td>
<td></td>
<td>MORElimit</td>
</tr>
<tr>
<td>MasterAddress</td>
<td></td>
<td>MOREThan</td>
</tr>
<tr>
<td>MasterRead</td>
<td></td>
<td>MOREthan</td>
</tr>
<tr>
<td>MasterWrite</td>
<td></td>
<td>MOSFET</td>
</tr>
<tr>
<td>Math</td>
<td></td>
<td>MSB</td>
</tr>
<tr>
<td>MathArbflt</td>
<td></td>
<td>MULTIply</td>
</tr>
<tr>
<td>MathFFTView</td>
<td></td>
<td>Multipleframes</td>
</tr>
<tr>
<td>MathFFTx</td>
<td></td>
<td>Mathx</td>
</tr>
<tr>
<td>Math</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>Mathx</td>
<td></td>
<td>NAK</td>
</tr>
<tr>
<td>Math</td>
<td></td>
<td>NAME</td>
</tr>
<tr>
<td>Max</td>
<td></td>
<td>NAME</td>
</tr>
<tr>
<td>MAXFRAMES</td>
<td></td>
<td>NAND</td>
</tr>
<tr>
<td>Maximum</td>
<td></td>
<td>NAND</td>
</tr>
<tr>
<td>MAXSamplerate</td>
<td></td>
<td>Navigate</td>
</tr>
<tr>
<td>Maximum</td>
<td></td>
<td>NDUTYCYCLE</td>
</tr>
<tr>
<td>MDATA</td>
<td></td>
<td>NDUTY</td>
</tr>
<tr>
<td>ME</td>
<td></td>
<td>NDuty</td>
</tr>
<tr>
<td>MEAN</td>
<td></td>
<td>NEGative</td>
</tr>
<tr>
<td>MEANAUTOCalculate</td>
<td></td>
<td>NEGative</td>
</tr>
<tr>
<td>MEANhistogram</td>
<td></td>
<td>NETWORKConfig</td>
</tr>
<tr>
<td>MEASName</td>
<td></td>
<td>NEWpass</td>
</tr>
<tr>
<td>MEASRange</td>
<td></td>
<td>NEXT</td>
</tr>
<tr>
<td>MEASurement</td>
<td></td>
<td>NEXT</td>
</tr>
<tr>
<td>MEAS</td>
<td></td>
<td>NIBBLECount</td>
</tr>
<tr>
<td>MEASx</td>
<td></td>
<td>NOCARE</td>
</tr>
<tr>
<td>Median</td>
<td></td>
<td>NOISEAdd</td>
</tr>
<tr>
<td>Medium</td>
<td></td>
<td>NOISerj</td>
</tr>
<tr>
<td>MEMORY</td>
<td></td>
<td>NOISE</td>
</tr>
<tr>
<td>Method</td>
<td></td>
<td>NOMINALOFFSET</td>
</tr>
<tr>
<td>MHT</td>
<td></td>
<td>NOMinal</td>
</tr>
<tr>
<td>MID</td>
<td></td>
<td>NOMinalfreq</td>
</tr>
<tr>
<td>MIDRef</td>
<td></td>
<td>None</td>
</tr>
<tr>
<td>MIL</td>
<td></td>
<td>NONTRANSition</td>
</tr>
<tr>
<td>MIL1553B</td>
<td></td>
<td>None</td>
</tr>
<tr>
<td>MIN</td>
<td></td>
<td>NOPARity</td>
</tr>
<tr>
<td>MINIMUM</td>
<td></td>
<td>NOR</td>
</tr>
<tr>
<td>MINimum</td>
<td></td>
<td>NRESPOnse</td>
</tr>
<tr>
<td>MINMax</td>
<td></td>
<td>Normal</td>
</tr>
<tr>
<td>MINuI</td>
<td></td>
<td>Normal</td>
</tr>
<tr>
<td>Minimum</td>
<td></td>
<td>NOTE</td>
</tr>
<tr>
<td>MIXED</td>
<td></td>
<td>NOVERSHOOT</td>
</tr>
<tr>
<td>MIXEDASCII</td>
<td></td>
<td>None</td>
</tr>
<tr>
<td>MIXEDHEX</td>
<td></td>
<td>NPERIOD</td>
</tr>
<tr>
<td>MIXed</td>
<td></td>
<td>NPulseWidth</td>
</tr>
<tr>
<td>MKDir</td>
<td></td>
<td>Output5</td>
</tr>
<tr>
<td>MODE</td>
<td></td>
<td>Output5</td>
</tr>
<tr>
<td>MODEhistogram</td>
<td></td>
<td>Output6</td>
</tr>
<tr>
<td>MODE</td>
<td></td>
<td>Output7</td>
</tr>
<tr>
<td>NR1</td>
<td></td>
<td>OUTside</td>
</tr>
<tr>
<td>NR2</td>
<td></td>
<td>Output5</td>
</tr>
<tr>
<td>NR3</td>
<td></td>
<td>OUTside</td>
</tr>
<tr>
<td>OUTPUT</td>
<td></td>
<td>Override</td>
</tr>
<tr>
<td>OVERRIDE</td>
<td></td>
<td>OVERlayment</td>
</tr>
<tr>
<td>OVERlayment</td>
<td></td>
<td>Output2Pwr</td>
</tr>
<tr>
<td>OVERload</td>
<td></td>
<td>Output3Pwr</td>
</tr>
<tr>
<td>OVER</td>
<td></td>
<td>OutputPwr</td>
</tr>
<tr>
<td>Word</td>
<td>Word</td>
<td>Word</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td>PLL</td>
<td>QFACTOR</td>
<td>RATE9K</td>
</tr>
<tr>
<td>PLOT</td>
<td>QString</td>
<td>RATE9M</td>
</tr>
<tr>
<td>PLOTData</td>
<td>QStringNR1</td>
<td>RATE ded</td>
</tr>
<tr>
<td>PLOTIMAGE</td>
<td>QStringQString</td>
<td>RCURRENT</td>
</tr>
<tr>
<td>PLOTVIEW</td>
<td>QTAG</td>
<td>RDS</td>
</tr>
<tr>
<td>PLOTView</td>
<td>QTAGGING</td>
<td>RDSOn</td>
</tr>
<tr>
<td>PLOTx</td>
<td>Qstring</td>
<td>READ</td>
</tr>
<tr>
<td>PNG</td>
<td>RADIans</td>
<td>REAL</td>
</tr>
<tr>
<td>PNJUNCTION</td>
<td>RAMP</td>
<td>RECALL</td>
</tr>
<tr>
<td>POHCL</td>
<td>RAMPtime</td>
<td>RECALLmask</td>
</tr>
<tr>
<td>POHCM</td>
<td>RANDOM</td>
<td>RECORD</td>
</tr>
<tr>
<td>POHCS</td>
<td>RATE100K</td>
<td>RECORDLength</td>
</tr>
<tr>
<td>POIN</td>
<td>RATE10K</td>
<td>RECORDlength</td>
</tr>
<tr>
<td>POLARITY</td>
<td>RATE10M</td>
<td>RECTANGular</td>
</tr>
<tr>
<td>POLarity</td>
<td>RATE115K</td>
<td>REDO</td>
</tr>
<tr>
<td>POPulation</td>
<td>RATE11M</td>
<td>REF</td>
</tr>
<tr>
<td>PORT</td>
<td>RATE125K</td>
<td>REFFFTView</td>
</tr>
<tr>
<td>POSITIVE</td>
<td>RATE12M</td>
<td>REFFFTx</td>
</tr>
<tr>
<td>POST</td>
<td>RATE13M</td>
<td>REFLevel</td>
</tr>
<tr>
<td>Position</td>
<td>RATE14M</td>
<td>REFLevels</td>
</tr>
<tr>
<td>Positive</td>
<td>RATE153K</td>
<td>REFLevels</td>
</tr>
<tr>
<td>POVERSHOOT</td>
<td>RATE15M</td>
<td>RFMODE</td>
</tr>
<tr>
<td>POWERQUALITY</td>
<td>RATE16M</td>
<td>REFOUT</td>
</tr>
<tr>
<td>POWERrating</td>
<td>RATE19K</td>
<td>REFVoltage</td>
</tr>
<tr>
<td>POWERTABLE</td>
<td>RATE1K</td>
<td>Reference</td>
</tr>
<tr>
<td>Power</td>
<td>RATE1M</td>
<td>REF</td>
</tr>
<tr>
<td>Power</td>
<td>RATE20K</td>
<td>REF_D</td>
</tr>
<tr>
<td>PPULSEWIDTH</td>
<td>RATE250K</td>
<td>REF_D</td>
</tr>
<tr>
<td>PRESS</td>
<td>RATE25K</td>
<td>REFx</td>
</tr>
<tr>
<td>PREViewstate</td>
<td>RATE2K</td>
<td>REFx_DX</td>
</tr>
<tr>
<td>PREvious</td>
<td>RATE2M</td>
<td>REG0WRITE</td>
</tr>
<tr>
<td>PREamble</td>
<td>RATE300</td>
<td>REGISTERADDRESS</td>
</tr>
<tr>
<td>Prev</td>
<td>RATE31K</td>
<td>REGREAD</td>
</tr>
<tr>
<td>PRIMARYTURNS</td>
<td>RATE33K</td>
<td>REGWRIte</td>
</tr>
<tr>
<td>PROBECntrol</td>
<td>RATE38K</td>
<td>REM</td>
</tr>
<tr>
<td>PROBEcal</td>
<td>RATE3M</td>
<td>REMote</td>
</tr>
<tr>
<td>PROBEFunc</td>
<td>RATE400K</td>
<td>RENAME</td>
</tr>
<tr>
<td>Protocol</td>
<td>RATE4K</td>
<td>REPEATstart</td>
</tr>
<tr>
<td>PROTOCOL</td>
<td>RATE4M</td>
<td>REPORT</td>
</tr>
<tr>
<td>Probe</td>
<td>RATE500K</td>
<td>REPWR</td>
</tr>
<tr>
<td>PT_FMT</td>
<td>RATE50K</td>
<td>REPeating</td>
</tr>
<tr>
<td>PT.getOrder</td>
<td>RATE5M</td>
<td>RESERVED</td>
</tr>
<tr>
<td>PT_off</td>
<td>RATE62K</td>
<td>RESET</td>
</tr>
<tr>
<td>PULSEWidth</td>
<td>RATE68K</td>
<td>RESET</td>
</tr>
<tr>
<td>PULSE</td>
<td>RATE6M</td>
<td>Resolution</td>
</tr>
<tr>
<td>PULSE</td>
<td>RATE7M</td>
<td>RESponsetime</td>
</tr>
<tr>
<td>WIDTH</td>
<td>RATE800K</td>
<td>RESUME</td>
</tr>
<tr>
<td>PWRFACtor</td>
<td>RATE83K</td>
<td>RESULT</td>
</tr>
<tr>
<td>PWRFREQ</td>
<td>RATE8M</td>
<td>RESULts</td>
</tr>
<tr>
<td>PWRUpstatus</td>
<td>RATE921K</td>
<td>RESistance</td>
</tr>
<tr>
<td>Permeability</td>
<td>RATE92K</td>
<td>RESolution</td>
</tr>
<tr>
<td>Reserved Word</td>
<td>Reserved Word</td>
<td>Reserved Word</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>SC</td>
<td>SCALE</td>
<td>SCALERatio</td>
</tr>
<tr>
<td>SCALE</td>
<td>SDF</td>
<td>SHORT</td>
</tr>
<tr>
<td>SCIENTIFIC</td>
<td>SIGNAL</td>
<td>SIGNALTYPE</td>
</tr>
<tr>
<td>SCREEN</td>
<td>SI</td>
<td>SI</td>
</tr>
<tr>
<td>SDI</td>
<td>SDIDATA</td>
<td>SDIDATAASSM</td>
</tr>
<tr>
<td>SEARCH</td>
<td>SEARCH1</td>
<td>SEARCHSOURCE</td>
</tr>
<tr>
<td>SEARCHTABLE</td>
<td>SEARCHHITOTRIGGER</td>
<td>SEARCHx</td>
</tr>
<tr>
<td>SEC</td>
<td>SEC1SOURCE</td>
<td>SEC1TURNS</td>
</tr>
<tr>
<td>SEC3SOURCE</td>
<td>SEC3TURNS</td>
<td>SEC4SOURCE</td>
</tr>
<tr>
<td>SEC5SOURCE</td>
<td>SEC5TURNS</td>
<td>SEC6SOURCE</td>
</tr>
<tr>
<td>SECOND</td>
<td>SLOPE</td>
<td>SLOPE</td>
</tr>
<tr>
<td>SECONDS</td>
<td>SMPS</td>
<td>SNAp</td>
</tr>
<tr>
<td>SECWINDINGS</td>
<td>SELECTED</td>
<td>SELECTED</td>
</tr>
<tr>
<td>SELECTED</td>
<td>SOFFRAMENUMBER</td>
<td>SOLID</td>
</tr>
<tr>
<td>SENT</td>
<td>SEQNUM</td>
<td>SEQUENCE</td>
</tr>
<tr>
<td>SEQNUM</td>
<td>SOURCE</td>
<td>SOURCEEDGEType</td>
</tr>
<tr>
<td>SERVICE</td>
<td>SOURCELIST</td>
<td>SOURCEMODeA</td>
</tr>
<tr>
<td>SERVICESNAME</td>
<td>SOURCEnumber</td>
<td>SESSION</td>
</tr>
<tr>
<td>SESSION</td>
<td>SOURCES</td>
<td>SOURCES</td>
</tr>
<tr>
<td>SOURCE</td>
<td>SOURCES</td>
<td>SOURCE</td>
</tr>
<tr>
<td>SOURCEDADDR</td>
<td>SOURCE</td>
<td>SOURCEADDR</td>
</tr>
<tr>
<td>SOURCEPORT</td>
<td>SOURCEPORT</td>
<td>SOURCETYPE</td>
</tr>
<tr>
<td>SOURCE</td>
<td>SOURCETYPE</td>
<td>SOURCETYPE</td>
</tr>
<tr>
<td>SOURCETYPE</td>
<td>SOURCETYPE</td>
<td>SOURCETYPE</td>
</tr>
<tr>
<td>SOURCETYPE</td>
<td>SOURCETYPE</td>
<td>SOURCETYPE</td>
</tr>
<tr>
<td>SOURCETYPE</td>
<td>SOURCETYPE</td>
<td>SOURCETYPE</td>
</tr>
<tr>
<td>SOURCETYPE</td>
<td>SOURCETYPE</td>
<td>SOURCETYPE</td>
</tr>
<tr>
<td>SOURCETYPE</td>
<td>SOURCETYPE</td>
<td>SOURCETYPE</td>
</tr>
<tr>
<td>Reserved Words</td>
<td>Reserved Words</td>
<td>Reserved Words</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>----------------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>TOEdge</td>
<td>TX</td>
<td>VDIFFXOVR</td>
</tr>
<tr>
<td>TOFFENRG</td>
<td>TXRX</td>
<td>VECTors</td>
</tr>
<tr>
<td>TOFFLOSS</td>
<td>TXRXTHRESHold</td>
<td>VEBOse</td>
</tr>
<tr>
<td>TOKENPacket</td>
<td>TYPE</td>
<td>VERTPOS</td>
</tr>
<tr>
<td>TOKENLOSS</td>
<td>TYPE1</td>
<td>VERTSCALE</td>
</tr>
<tr>
<td>TOKENType</td>
<td>TYPE2</td>
<td>VERTical</td>
</tr>
<tr>
<td>TONEnerg</td>
<td>TYPE</td>
<td>WORD</td>
</tr>
<tr>
<td>TONLOSS</td>
<td>TotalEfficiency</td>
<td>VIEW</td>
</tr>
<tr>
<td>TOP</td>
<td>UNDERline</td>
<td>WORDSIZE</td>
</tr>
<tr>
<td>TOTAL</td>
<td>UNDO</td>
<td>WRITE</td>
</tr>
<tr>
<td>TOTALuptime</td>
<td>UNEqual</td>
<td>X</td>
</tr>
<tr>
<td>TOUCHSCreen</td>
<td>UNINSTALL</td>
<td>XAUI</td>
</tr>
<tr>
<td>TRACK</td>
<td>UNIQUE</td>
<td>XAUI_GEN2</td>
</tr>
<tr>
<td>TRANSferbusownership</td>
<td>UNITINTERVAL</td>
<td>VOLTAGE</td>
</tr>
<tr>
<td>TRANSition</td>
<td>UNITIntervals</td>
<td>VOLTage</td>
</tr>
<tr>
<td>TRANSition</td>
<td>UNITInterval</td>
<td>VAXISUnits</td>
</tr>
<tr>
<td>TRBit</td>
<td>UNITS</td>
<td>XFF</td>
</tr>
<tr>
<td>TREND</td>
<td>UNIts</td>
<td>XINcr</td>
</tr>
<tr>
<td>TREND</td>
<td>UNLock</td>
<td>XUNIT</td>
</tr>
<tr>
<td>TRIGLevel</td>
<td>UNMOUNT</td>
<td>XY</td>
</tr>
<tr>
<td>TRIGMode</td>
<td>WRap</td>
<td>XYZ</td>
</tr>
<tr>
<td>TRIGslope</td>
<td>UPPERFREQuency</td>
<td>XUNIT</td>
</tr>
<tr>
<td>TRIGGER</td>
<td>Upper</td>
<td>Y</td>
</tr>
<tr>
<td>TRIGGERtosearch</td>
<td>UPPERthreshold</td>
<td>YAXIS</td>
</tr>
<tr>
<td>TRPWR</td>
<td>USB</td>
<td>YES</td>
</tr>
<tr>
<td>TRUEPWR</td>
<td>USB3</td>
<td>YMUlt</td>
</tr>
<tr>
<td>TRUE</td>
<td>USBDevice</td>
<td>YOFF</td>
</tr>
<tr>
<td>TTL</td>
<td>USBTmc</td>
<td>YPOS</td>
</tr>
<tr>
<td>TTLENRG</td>
<td>USEClockedge</td>
<td>YUNIt</td>
</tr>
<tr>
<td>TLLoss</td>
<td>USEClockedge</td>
<td>YZEO</td>
</tr>
<tr>
<td>TURN</td>
<td>USER</td>
<td>ZERo</td>
</tr>
<tr>
<td>TURNOFFtime</td>
<td>UTCDELTa</td>
<td>ZONE</td>
</tr>
<tr>
<td>TURNONtime</td>
<td>V1X</td>
<td>ZOOM</td>
</tr>
<tr>
<td>TWELVEtwelve</td>
<td>V2X</td>
<td>ZOOMNR1</td>
</tr>
<tr>
<td>TWENTyeighty</td>
<td>VALidate</td>
<td>ZOOMMOVERide</td>
</tr>
<tr>
<td>TWENTyfive</td>
<td>VALue</td>
<td>ZOOM</td>
</tr>
<tr>
<td>TWO</td>
<td>VARPersist</td>
<td>Zoom</td>
</tr>
<tr>
<td>TWOOCLOCKCYCLES</td>
<td>VBars</td>
<td>WIDTH</td>
</tr>
<tr>
<td>TWOFifty</td>
<td>VCESat</td>
<td>WIDTHTHBER</td>
</tr>
<tr>
<td>TWOThousand</td>
<td>VCFACtor</td>
<td>WIDh</td>
</tr>
</tbody>
</table>

Appendix B: Reserved Words
**Appendix C: Factory Defaults**

**Default Setup**

The following table lists the default values for each command.

*NOTE.* Find the most up-to-date default values for your instrument and software by performing a TekSecure command, saving the instrument setup and looking at the instrument or setup file.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQUIRE:FASTACQ:PALETTE TEMPERATURE</td>
<td></td>
</tr>
<tr>
<td>ACQUIRE:FASTACQ:STATE 0</td>
<td></td>
</tr>
<tr>
<td>ACQUIRE:MAGNIVU 0</td>
<td></td>
</tr>
<tr>
<td>ACQUIRE:MODE SAMPLE</td>
<td></td>
</tr>
<tr>
<td>ACQUIRE:NUMAVG 16</td>
<td></td>
</tr>
<tr>
<td>ACQUIRE:NUMENV INFINITE</td>
<td></td>
</tr>
<tr>
<td>ACQUIRE:STATE 1</td>
<td></td>
</tr>
<tr>
<td>ACQUIRE:STOPAFTER RUNSTOP</td>
<td></td>
</tr>
<tr>
<td>ACTONEVENT:ACTION:AUXOUT:STATE 0</td>
<td></td>
</tr>
<tr>
<td>ACTONEVENT:ACTION:EMAIL:SETUP:TOADDRESS &quot;TestString&quot;</td>
<td></td>
</tr>
<tr>
<td>ACTONEVENT:ACTION:EMAIL:STATE 0</td>
<td></td>
</tr>
<tr>
<td>ACTONEVENT:ACTION:PRINT:STATE 0</td>
<td></td>
</tr>
<tr>
<td>ACTONEVENT:ACTION:SAVEIMAGE:STATE 0</td>
<td></td>
</tr>
<tr>
<td>ACTONEVENT:ACTION:SAVEWFM:STATE 0</td>
<td></td>
</tr>
<tr>
<td>ACTONEVENT:ACTION:SRQ:STATE 0</td>
<td></td>
</tr>
<tr>
<td>ACTONEVENT:ACTION:STOPACQ:STATE 1</td>
<td></td>
</tr>
<tr>
<td>ACTONEVENT:ACTION:VISUAL:STATE 0</td>
<td></td>
</tr>
<tr>
<td>ACTONEVENT:EVENTTYPE NONE</td>
<td></td>
</tr>
<tr>
<td>ACTONEVENT:NUMACQS 1</td>
<td></td>
</tr>
<tr>
<td>ACTONEVENT:REPEATCOUNT 1.0000</td>
<td></td>
</tr>
<tr>
<td>AFG:AMPLITUDE 500.0000E-3</td>
<td></td>
</tr>
<tr>
<td>AFG:ARBITRARY:EMEM:POINTS:ENCDG ASCII</td>
<td></td>
</tr>
<tr>
<td>AFG:FREQUENCY 100.0E+3</td>
<td></td>
</tr>
<tr>
<td>AFG:FUNCTION SINE</td>
<td></td>
</tr>
<tr>
<td>AFG:HIGHLEVEL 250.0000E-3</td>
<td></td>
</tr>
<tr>
<td>AFG:LEVELPRESET USER</td>
<td></td>
</tr>
<tr>
<td>AFG:LOWLEVEL -250.0000E-3</td>
<td></td>
</tr>
</tbody>
</table>
### Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFG: NOISEADD: PERCENT</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>AFG: NOISEADD: STATE</td>
<td>0</td>
</tr>
<tr>
<td>AFG: OFFSET</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>AFG: OUTPUT: LOAD: IMPEDANCE</td>
<td>HIGHZ</td>
</tr>
<tr>
<td>AFG: PERIOD</td>
<td>9.9999999999916E-6</td>
</tr>
<tr>
<td>AFG: PHASE</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>AFG: PULSE: WIDTH</td>
<td>1.0E-6</td>
</tr>
<tr>
<td>AFG: RAMP: SYMMETRY</td>
<td>50.0000</td>
</tr>
<tr>
<td>AFG: SQUARE: DUTY</td>
<td>50.0000</td>
</tr>
<tr>
<td>ALIAS: STATE</td>
<td>0</td>
</tr>
<tr>
<td>APPLICATION: TYPE</td>
<td>POWER</td>
</tr>
<tr>
<td>AUXOUT: EDGE</td>
<td>RISING</td>
</tr>
<tr>
<td>AUXOUT: SOURCE</td>
<td>ATRIGGER</td>
</tr>
<tr>
<td>BUS:B1: AUDIO: BITDELAY</td>
<td>1</td>
</tr>
<tr>
<td>BUS:B1: AUDIO: BITORDER</td>
<td>MSB</td>
</tr>
<tr>
<td>BUS:B1: AUDIO: CHANNEL: SIZE</td>
<td>24</td>
</tr>
<tr>
<td>BUS:B1: AUDIO: CLOCK: POLARITY</td>
<td>RISE</td>
</tr>
<tr>
<td>BUS:B1: AUDIO: CLOCK: SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B1: AUDIO: DATA: POLARITY</td>
<td>NORMAL</td>
</tr>
<tr>
<td>BUS:B1: AUDIO: DATA: SIZE</td>
<td>24</td>
</tr>
<tr>
<td>BUS:B1: AUDIO: DISPLAY: FORMAT</td>
<td>SIGNEDDECIMAL</td>
</tr>
<tr>
<td>BUS:B1: AUDIO: FRAME: SIZE</td>
<td>8</td>
</tr>
<tr>
<td>BUS:B1: AUDIO: FRAMESYNC: POLARITY</td>
<td>RISE</td>
</tr>
<tr>
<td>BUS:B1: AUDIO: FRAMESYNC: SOURCE</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B1: AUDIO: TYPE</td>
<td>I2S</td>
</tr>
<tr>
<td>BUS:B1: AUDIO: WORDSEL: POLARITY</td>
<td>NORMAL</td>
</tr>
<tr>
<td>BUS:B1: AUDIO: WORDSEL: SOURCE</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B1: CAN: BITRATE</td>
<td>500000</td>
</tr>
<tr>
<td>BUS:B1: CAN: FD: BITRATE</td>
<td>4000000</td>
</tr>
<tr>
<td>BUS:B1: CAN: PROBE</td>
<td>CANH</td>
</tr>
<tr>
<td>BUS:B1: CAN: SAMPLEPOINT</td>
<td>50</td>
</tr>
<tr>
<td>BUS:B1: CAN: SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B1: CAN: STANDARD</td>
<td>CAN2X</td>
</tr>
<tr>
<td>BUS:B1: DISPLAY: FORMAT</td>
<td>HEXADECIMAL</td>
</tr>
<tr>
<td>BUS:B1: DISPLAY: TYPE</td>
<td>BUS</td>
</tr>
<tr>
<td>BUS:B1: FLEXRAY: BITRATE</td>
<td>10000000</td>
</tr>
</tbody>
</table>
### Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS:B1:FLEXRAY:CHANNEL A</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:FLEXRAY:SIGNAL BDIFFBP</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:FLEXRAY:SOURCE CH1</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:I2C:ADDRESS:RWINCLUDE 0</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:I2C:CLOCK:SOURCE CH1</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:I2C:DATA:SOURCE CH2</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:LABEL &quot;Parallel&quot;</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:LIN:BITRATE 19200</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:LIN:IDFORMAT NOPARITY</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:LIN:POLARITY NORMAL</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:LIN:SAMPLEPOINT 50</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:LIN:SOURCE CH1</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:LIN:STANDARD V2X</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:MIL1553B:POLARITY NORMAL</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:MIL1553B:RESPONSETIME:MAXIMUM 12.0000E-6</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:MIL1553B:RESPONSETIME:MINIMUM 4.0000E-6</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:MIL1553B:SOURCE CH1</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:BIT0:SOURCE D0</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:BIT14:SOURCE D14</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:BIT15:SOURCE D15</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:BIT16:SOURCE CH1</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:BIT17:SOURCE CH2</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:BIT18:SOURCE CH3</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:BIT19:SOURCE CH4</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:BIT1:SOURCE D1</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:BIT2:SOURCE D2</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:BIT4:SOURCE D4</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:BIT5:SOURCE D5</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:BIT6:SOURCE D6</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:BIT7:SOURCE D7</td>
<td></td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:BIT8:SOURCE D8</td>
<td></td>
</tr>
</tbody>
</table>
## Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS:B1:PARALLEL:CLOCK:EDGE</td>
<td>RISING</td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:CLOCK:ISLOCKED</td>
<td>NO</td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:CLOCK:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B1:PARALLEL:WIDTH</td>
<td>16</td>
</tr>
<tr>
<td>BUS:B1:POSITION</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:B1:RS232C:BITRATE</td>
<td>9600</td>
</tr>
<tr>
<td>BUS:B1:RS232C:DATABITS</td>
<td>8</td>
</tr>
<tr>
<td>BUS:B1:RS232C:DELIMITER</td>
<td>LF</td>
</tr>
<tr>
<td>BUS:B1:RS232C:DISPLAYMODE</td>
<td>FRAME</td>
</tr>
<tr>
<td>BUS:B1:RS232C:PARITY</td>
<td>NONE</td>
</tr>
<tr>
<td>BUS:B1:RS232C:POLARITY</td>
<td>NORMAL</td>
</tr>
<tr>
<td>BUS:B1:RS232C:RX:SOURCE</td>
<td>OFF</td>
</tr>
<tr>
<td>BUS:B1:RS232C:TX:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B1:SPI:BITORDER</td>
<td>MSB</td>
</tr>
<tr>
<td>BUS:B1:SPI:CLOCK:POLARITY</td>
<td>RISE</td>
</tr>
<tr>
<td>BUS:B1:SPI:CLOCK:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B1:SPI:DATA:IN:POLARITY</td>
<td>HIGH</td>
</tr>
<tr>
<td>BUS:B1:SPI:DATA:IN:SOURCE</td>
<td>OFF</td>
</tr>
<tr>
<td>BUS:B1:SPI:DATA:OUT:POLARITY</td>
<td>HIGH</td>
</tr>
<tr>
<td>BUS:B1:SPI:DATA:OUT:SOURCE</td>
<td>CH3</td>
</tr>
<tr>
<td>BUS:B1:SPI:DATA:SIZE</td>
<td>8</td>
</tr>
<tr>
<td>BUS:B1:SPI:FRAMING</td>
<td>SS</td>
</tr>
<tr>
<td>BUS:B1:SPI:IDLETIME</td>
<td>5.0000E-6</td>
</tr>
<tr>
<td>BUS:B1:SPI:SELECT:POLARITY</td>
<td>LOW</td>
</tr>
<tr>
<td>BUS:B1:SPI:SELECT:SOURCE</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B1:STATE</td>
<td>0</td>
</tr>
<tr>
<td>BUS:B1:TYPE</td>
<td>PARALLEL</td>
</tr>
<tr>
<td>BUS:B1:USB:BITRATE</td>
<td>FULL</td>
</tr>
<tr>
<td>BUS:B1:USB:PROBE</td>
<td>DIFFERENTIAL</td>
</tr>
<tr>
<td>BUS:B1:USB:SOURCE:DMINUS</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B1:USB:SOURCE:DPLUS</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:AUDIO:BITDELAY</td>
<td>1</td>
</tr>
<tr>
<td>BUS:B2:AUDIO:BITORDER</td>
<td>MSB</td>
</tr>
<tr>
<td>BUS:B2:AUDIO:CLOCK:POLARITY</td>
<td>RISE</td>
</tr>
<tr>
<td>BUS:B2:AUDIO:CLOCK:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>BUS:B2:AUDIO:DATA:POLARITY</td>
<td>NORMAL</td>
</tr>
<tr>
<td>BUS:B2:AUDIO:DISPLAY:FORMAT</td>
<td>SIGNEDDECIMAL</td>
</tr>
<tr>
<td>BUS:B2:AUDIO:FRAME:SIZE</td>
<td>8</td>
</tr>
<tr>
<td>BUS:B2:AUDIO:FRAMESYNC:POLARITY</td>
<td>RISE</td>
</tr>
<tr>
<td>BUS:B2:AUDIO:FRAMESYNC:SOURCE</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B2:AUDIO:TYPE</td>
<td>I2S</td>
</tr>
<tr>
<td>BUS:B2:AUDIO:WORDSEL:POLARITY</td>
<td>NORMAL</td>
</tr>
<tr>
<td>BUS:B2:AUDIO:WORDSEL:SOURCE</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B2:CAN:BITRATE</td>
<td>500000</td>
</tr>
<tr>
<td>BUS:B2:CAN:FD:BITRATE</td>
<td>4000000</td>
</tr>
<tr>
<td>BUS:B2:CAN:PROBE</td>
<td>CANH</td>
</tr>
<tr>
<td>BUS:B2:CAN:SAMPLEPOINT</td>
<td>50</td>
</tr>
<tr>
<td>BUS:B2:CAN:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:CAN:STANDARD</td>
<td>CAN2X</td>
</tr>
<tr>
<td>BUS:B2:DISPLAY:FORMAT</td>
<td>HEXADECIMAL</td>
</tr>
<tr>
<td>BUS:B2:DISPLAY:TYPE</td>
<td>BUS</td>
</tr>
<tr>
<td>BUS:B2:FLEXRAY:BITRATE</td>
<td>10000000</td>
</tr>
<tr>
<td>BUS:B2:FLEXRAY:CHANNEL</td>
<td>A</td>
</tr>
<tr>
<td>BUS:B2:FLEXRAY:SIGNAL</td>
<td>BDIFFBP</td>
</tr>
<tr>
<td>BUS:B2:FLEXRAY:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:I2C:ADDRESS:RWINCLUDE</td>
<td>0</td>
</tr>
<tr>
<td>BUS:B2:I2C:CLOCK:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:I2C:DATA:SOURCE</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B2:LABEL</td>
<td>&quot;Parallel&quot;</td>
</tr>
<tr>
<td>BUS:B2:LIN:BITRATE</td>
<td>19200</td>
</tr>
<tr>
<td>BUS:B2:LIN:IDFORMAT</td>
<td>NOPARITY</td>
</tr>
<tr>
<td>BUS:B2:LIN:POLARITY</td>
<td>NORMAL</td>
</tr>
<tr>
<td>BUS:B2:LIN:SAMPLEPOINT</td>
<td>50</td>
</tr>
<tr>
<td>BUS:B2:LIN:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:LIN:STANDARD</td>
<td>V2X</td>
</tr>
<tr>
<td>BUS:B2:MIL1553B:POLARITY</td>
<td>NORMAL</td>
</tr>
<tr>
<td>BUS:B2:MIL1553B:RESPONSETIME:MINIMUM</td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>BUS:B2:MIL1553B:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:PARALLEL:BIT0:SOURCE</td>
<td>D0</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>BUS:B2:PARALLEL:BIT16:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:PARALLEL:BIT17:SOURCE</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B2:PARALLEL:BIT1:SOURCE</td>
<td>D1</td>
</tr>
<tr>
<td>BUS:B2:PARALLEL:BIT8:SOURCE</td>
<td>D8</td>
</tr>
<tr>
<td>BUS:B2:PARALLEL:CLOCK:EDGE</td>
<td>RISING</td>
</tr>
<tr>
<td>BUS:B2:PARALLEL:CLOCK:ISCLOCKED</td>
<td>NO</td>
</tr>
<tr>
<td>BUS:B2:PARALLEL:CLOCK:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:PARALLEL:WIDTH</td>
<td>16</td>
</tr>
<tr>
<td>BUS:B2:POSITION</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:B2:RS232C:BITRATE</td>
<td>9600</td>
</tr>
<tr>
<td>BUS:B2:RS232C:DATABITS</td>
<td>8</td>
</tr>
<tr>
<td>BUS:B2:RS232C:DELIMITER</td>
<td>LF</td>
</tr>
<tr>
<td>BUS:B2:RS232C:DISPLAYMODE</td>
<td>FRAME</td>
</tr>
<tr>
<td>BUS:B2:RS232C:PARITY</td>
<td>NONE</td>
</tr>
<tr>
<td>BUS:B2:RS232C:POLARITY</td>
<td>NORMAL</td>
</tr>
<tr>
<td>BUS:B2:RS232C:TX:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:SPI:BITORDER</td>
<td>MSB</td>
</tr>
<tr>
<td>BUS:B2:SPI:CLOCK:POLARITY</td>
<td>RISE</td>
</tr>
<tr>
<td>BUS:B2:SPI:CLOCK:SOURCE</td>
<td>CH1</td>
</tr>
</tbody>
</table>
### Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS:B2:SPI:DATA:SIZE</td>
<td>8</td>
</tr>
<tr>
<td>BUS:B2:SPI:FRAMING</td>
<td>SS</td>
</tr>
<tr>
<td>BUS:B2:SPI:IDLETIME</td>
<td>5.0000E-6</td>
</tr>
<tr>
<td>BUS:B2:SPI:SELECT:POLARITY</td>
<td>LOW</td>
</tr>
<tr>
<td>BUS:B2:SPI:SELECT:SOURCE</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B2:STATE</td>
<td>0</td>
</tr>
<tr>
<td>BUS:B2:TYPE</td>
<td>PARALLEL</td>
</tr>
<tr>
<td>BUS:B2:USB:BITRATE</td>
<td>FULL</td>
</tr>
<tr>
<td>BUS:B2:USB:PROBE</td>
<td>DIFFERENTIAL</td>
</tr>
<tr>
<td>BUS:B2:USB:SOURCE:DIFERENTIAL</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:USB:SOURCE:DPLUS</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:LOWER_THRESHOLD:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:LOWER_THRESHOLD:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:LOWER_THRESHOLD:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:LOWER_THRESHOLD:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:LOWER_THRESHOLD:MATH</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:LOWER_THRESHOLD:REF1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:LOWER_THRESHOLD:REF2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:LOWER_THRESHOLD:REF3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:LOWER_THRESHOLD:REF4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:UPPER_THRESHOLD:CH1</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>BUS:UPPER_THRESHOLD:CH2</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>BUS:UPPER_THRESHOLD:CH3</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>BUS:UPPER_THRESHOLD:CH4</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>BUS:UPPER_THRESHOLD:MATH</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:UPPER_THRESHOLD:REF1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:UPPER_THRESHOLD:REF2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:UPPER_THRESHOLD:REF3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:UPPER_THRESHOLD:REF4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH1:AMPSVIAVOLTS:ENABLE</td>
<td>0</td>
</tr>
<tr>
<td>CH1:AMPSVIAVOLTS:FACTOR</td>
<td>10.0000</td>
</tr>
<tr>
<td>CH1:BANDWIDTH</td>
<td>1.0000E+9</td>
</tr>
<tr>
<td>CH1:COUPLING</td>
<td>DC</td>
</tr>
<tr>
<td>CH1:DESKEW</td>
<td>0.0E+0</td>
</tr>
</tbody>
</table>
# Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH1:INVERT</td>
<td>0</td>
</tr>
<tr>
<td>CH1:LABEL</td>
<td>&quot;=&quot;</td>
</tr>
<tr>
<td>CH1:OFFSET</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH1:POSITION</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH1:PROBE:FORCEDRANGE</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH1:PROBE:GAIN</td>
<td>1.0000</td>
</tr>
<tr>
<td>CH1:PROBE:PROPDELAY</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH1:SCALE</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>CH1:TERMINATION</td>
<td>1.0000E+6</td>
</tr>
<tr>
<td>CH1:YUNITS</td>
<td>&quot;V&quot;</td>
</tr>
<tr>
<td>CH2:AMPSVIAVOLTS:ENABLE</td>
<td>0</td>
</tr>
<tr>
<td>CH2:AMPSVIAVOLTS:FACTOR</td>
<td>10.0000</td>
</tr>
<tr>
<td>CH2:BANDWIDTH</td>
<td>1.0000E+9</td>
</tr>
<tr>
<td>CH2:COUPLING</td>
<td>DC</td>
</tr>
<tr>
<td>CH2:DESKEW</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH2:INVERT</td>
<td>0</td>
</tr>
<tr>
<td>CH2:LABEL</td>
<td>&quot;=&quot;</td>
</tr>
<tr>
<td>CH2:OFFSET</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH2:POSITION</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH2:PROBE:FORCEDRANGE</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH2:PROBE:GAIN</td>
<td>1.0000</td>
</tr>
<tr>
<td>CH2:PROBE:PROPDELAY</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH2:SCALE</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>CH2:TERMINATION</td>
<td>1.0000E+6</td>
</tr>
<tr>
<td>CH2:YUNITS</td>
<td>&quot;V&quot;</td>
</tr>
<tr>
<td>CH3:AMPSVIAVOLTS:ENABLE</td>
<td>0</td>
</tr>
<tr>
<td>CH3:AMPSVIAVOLTS:FACTOR</td>
<td>10.0000</td>
</tr>
<tr>
<td>CH3:BANDWIDTH</td>
<td>1.0000E+9</td>
</tr>
<tr>
<td>CH3:COUPLING</td>
<td>DC</td>
</tr>
<tr>
<td>CH3:DESKEW</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH3:INVERT</td>
<td>0</td>
</tr>
<tr>
<td>CH3:LABEL</td>
<td>&quot;=&quot;</td>
</tr>
<tr>
<td>CH3:OFFSET</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH3:POSITION</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH3:PROBE:FORCEDRANGE</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH3:PROBE:GAIN</td>
<td>1.0000</td>
</tr>
<tr>
<td>CH3:PROBE:PROPDELAY</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>CH3:SCALE</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>CH3:TERMINATION</td>
<td>1.0000E+6</td>
</tr>
<tr>
<td>CH3:YUNITS</td>
<td>&quot;V&quot;</td>
</tr>
<tr>
<td>CH4:AMPSVIAVOLTS:ENABLE</td>
<td>0</td>
</tr>
<tr>
<td>CH4:AMPSVIAVOLTS:FACTOR</td>
<td>10.0000</td>
</tr>
<tr>
<td>CH4:BANDWIDTH</td>
<td>1.0000E+9</td>
</tr>
<tr>
<td>CH4:COUPLING</td>
<td>DC</td>
</tr>
<tr>
<td>CH4:DESKEW</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH4:INVERT</td>
<td>0</td>
</tr>
<tr>
<td>CH4:LABEL</td>
<td>**</td>
</tr>
<tr>
<td>CH4:OFFSET</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH4:POSITION</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH4:PROBE:FORCEDRANGE</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH4:PROBE:GAIN</td>
<td>1.0000</td>
</tr>
<tr>
<td>CH4:PROBE:PROPDELAY</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH4:SCALE</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>CH4:TERMINATION</td>
<td>1.0000E+6</td>
</tr>
<tr>
<td>CH4:YUNITS</td>
<td>&quot;V&quot;</td>
</tr>
<tr>
<td>CURSOR:FUNCTION</td>
<td>OFF</td>
</tr>
<tr>
<td>CURSOR:HBARS:POSITION1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CURSOR:HBARS:POSITION2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CURSOR:HBARS:UNITS</td>
<td>BASE</td>
</tr>
<tr>
<td>CURSOR:MODE</td>
<td>INDEPENDENT</td>
</tr>
<tr>
<td>CURSOR:SOURCE</td>
<td>AUTO</td>
</tr>
<tr>
<td>CURSOR:VBARS:POSITION1</td>
<td>-12.00E-6</td>
</tr>
<tr>
<td>CURSOR:VBARS:POSITION2</td>
<td>12.00E-6</td>
</tr>
<tr>
<td>CURSOR:VBARS:UNITS</td>
<td>SECONDS</td>
</tr>
<tr>
<td>CURSOR:XY:READOUT</td>
<td>RECTANGULAR</td>
</tr>
<tr>
<td>CURSOR:XY:RECTANGULAR:X:POSITION1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CURSOR:XY:RECTANGULAR:X:POSITION2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CURSOR:XY:RECTANGULAR:Y:POSITION1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CURSOR:XY:RECTANGULAR:Y:POSITION2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>D0:LABEL</td>
<td>**</td>
</tr>
<tr>
<td>D0:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D0:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>D10:LABEL</td>
<td>**</td>
</tr>
<tr>
<td>D10:POSITION</td>
<td>80.0000E-3</td>
</tr>
</tbody>
</table>
## Appendix C: Factory Defaults

Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D10:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>D11:LABEL</td>
<td></td>
</tr>
<tr>
<td>D11:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D11:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>D12:LABEL</td>
<td></td>
</tr>
<tr>
<td>D12:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D12:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>D13:LABEL</td>
<td></td>
</tr>
<tr>
<td>D13:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D13:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>D14:LABEL</td>
<td></td>
</tr>
<tr>
<td>D14:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D14:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>D15:LABEL</td>
<td></td>
</tr>
<tr>
<td>D15:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D15:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>D1:LABEL</td>
<td></td>
</tr>
<tr>
<td>D1:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D1:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>D2:LABEL</td>
<td></td>
</tr>
<tr>
<td>D2:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D2:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>D3:LABEL</td>
<td></td>
</tr>
<tr>
<td>D3:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D3:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>D4:LABEL</td>
<td></td>
</tr>
<tr>
<td>D4:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D4:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>D5:LABEL</td>
<td></td>
</tr>
<tr>
<td>D5:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D5:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>D6:LABEL</td>
<td></td>
</tr>
<tr>
<td>D6:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D6:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>D7:LABEL</td>
<td></td>
</tr>
<tr>
<td>D7:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D7:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>D8:LABEL</td>
<td></td>
</tr>
<tr>
<td>D8:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D8:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>D9:LABEL</td>
<td></td>
</tr>
<tr>
<td>D9:POSITION</td>
<td>80.0000E-3</td>
</tr>
<tr>
<td>D9:THRESHOLD</td>
<td>1.4000</td>
</tr>
<tr>
<td>DATA:DESTINATION</td>
<td>REF1</td>
</tr>
<tr>
<td>DATA:ENCDBG</td>
<td>RIBINARY</td>
</tr>
<tr>
<td>DATA:SOURC</td>
<td>CH1</td>
</tr>
<tr>
<td>DATA:START</td>
<td>1</td>
</tr>
<tr>
<td>DATA:STOP</td>
<td>10000</td>
</tr>
<tr>
<td>DESKEW:DISPLAY</td>
<td>1</td>
</tr>
<tr>
<td>DISPLAY:CLOCK</td>
<td>1</td>
</tr>
<tr>
<td>DISPLAY:DIGITAL:ACTIVITY</td>
<td>0</td>
</tr>
<tr>
<td>DISPLAY:DIGITAL:HEIGHT</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>DISPLAY:GRATICULE</td>
<td>FULL</td>
</tr>
<tr>
<td>DISPLAY:INTENSITY:BACKLIGHT</td>
<td>HIGH</td>
</tr>
<tr>
<td>DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE</td>
<td>1</td>
</tr>
<tr>
<td>DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:TIME</td>
<td>60</td>
</tr>
<tr>
<td>DISPLAY:INTENSITY:GRATICULE</td>
<td>75</td>
</tr>
<tr>
<td>DISPLAY:INTENSITY:WAVEFORM</td>
<td>35</td>
</tr>
<tr>
<td>DISPLAY:PERSISTENCE</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>DISPLAY:STYLE:DOTSONLY</td>
<td>0</td>
</tr>
<tr>
<td>DISPLAY:TRIGFREQUENCY</td>
<td>0</td>
</tr>
<tr>
<td>DISPLAY:XY</td>
<td>OFF</td>
</tr>
<tr>
<td>DISPLAY:XY:WITHYT</td>
<td>0</td>
</tr>
<tr>
<td>DVM:AUTORANGE</td>
<td>0</td>
</tr>
<tr>
<td>DVM:DISPLAYSTYLE</td>
<td>FULL</td>
</tr>
<tr>
<td>DVM:MODE</td>
<td>OFF</td>
</tr>
<tr>
<td>DVM:SOURC</td>
<td>CH1</td>
</tr>
<tr>
<td>EMAIL:SETUP:FROMADDRESS</td>
<td>&quot;TestString&quot;</td>
</tr>
<tr>
<td>EMAIL:SETUP:HOSTALIASNAME</td>
<td>&quot;TestString&quot;</td>
</tr>
<tr>
<td>EMAIL:SETUP:SMTLOGIN</td>
<td>&quot;TestString&quot;</td>
</tr>
<tr>
<td>EMAIL:SETUP:SMTPPORT</td>
<td>25</td>
</tr>
<tr>
<td>EMAIL:SETUP:SMTPSERVER</td>
<td>&quot;TestString&quot;</td>
</tr>
<tr>
<td>FASTACQ:STATE</td>
<td>0</td>
</tr>
<tr>
<td>HEADER</td>
<td>0</td>
</tr>
</tbody>
</table>
Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HISTOGRAM:BOXPCNT</td>
<td>20.0000, 20.0000, 80.0000, 80.0000</td>
</tr>
<tr>
<td>HISTOGRAM:DISPLAY</td>
<td>LINEAR</td>
</tr>
<tr>
<td>HISTOGRAM:MODE</td>
<td>OFF</td>
</tr>
<tr>
<td>HISTOGRAM:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>HORIZONTAL:DELAY:MODE</td>
<td>1</td>
</tr>
<tr>
<td>HORIZONTAL:DELAY:TIME</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>HORIZONTAL:POSITION</td>
<td>50.0000</td>
</tr>
<tr>
<td>HORIZONTAL:RECORDLENGTH</td>
<td>10000</td>
</tr>
<tr>
<td></td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>LOCK</td>
<td>NONE</td>
</tr>
<tr>
<td>MARKER:M1:FREQUENCY:ABSOLUTE</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>MARKER:M2:FREQUENCY:ABSOLUTE</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>MARKER:MANUAL</td>
<td>0</td>
</tr>
<tr>
<td>MARKER:PEAK:EXCURSION</td>
<td>30.0000</td>
</tr>
<tr>
<td>MARKER:PEAK:MAXIMUM</td>
<td>5</td>
</tr>
<tr>
<td>MARKER:PEAK:STATE</td>
<td>1</td>
</tr>
<tr>
<td>MARKER:PEAK:THRESHOLD</td>
<td>-50.0000</td>
</tr>
<tr>
<td>MARKER:TYPE</td>
<td>ABSOLUTE</td>
</tr>
<tr>
<td>MASK:TEST:SAVEWFM</td>
<td>0</td>
</tr>
<tr>
<td>MASK:TEST:SRQ:COMPLETION</td>
<td>0</td>
</tr>
<tr>
<td>MASK:TEST:SRQ:FAILURE</td>
<td>0</td>
</tr>
<tr>
<td>MASK:TEST:STATE</td>
<td>0</td>
</tr>
<tr>
<td>MASK:TEST:STOP:FAILURE</td>
<td>0</td>
</tr>
<tr>
<td>MASK:TEST:THRESHOLD</td>
<td>1</td>
</tr>
<tr>
<td>MASK:TEST:TIME</td>
<td>INFINITE</td>
</tr>
<tr>
<td>MASK:TEST:WAVEFORM</td>
<td>INFINITE</td>
</tr>
<tr>
<td>MASK:USER:AMPLITUDE</td>
<td>1.0000</td>
</tr>
<tr>
<td>MASK:USER:HSCALE</td>
<td>400.0000E-6</td>
</tr>
<tr>
<td>MASK:USER:HTRIGPOS</td>
<td>500.0000E-3</td>
</tr>
<tr>
<td>MATH:AUTOSCALE</td>
<td>1</td>
</tr>
<tr>
<td>MATH:DEFINE</td>
<td>&quot;CH1+CH2&quot;</td>
</tr>
<tr>
<td>MATH:HORIZONTAL:POSITION</td>
<td>50.0000</td>
</tr>
<tr>
<td>MATH:</td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>MATH:HORIZONTAL:UNITS</td>
<td>&quot;s&quot;</td>
</tr>
<tr>
<td>MATH:LABEL</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>MATH:SPECTRAL:MAG</td>
<td>DB</td>
</tr>
<tr>
<td>MATH:SPECTRAL:WINDOW</td>
<td>HANNING</td>
</tr>
</tbody>
</table>
Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH:TYPE</td>
<td>DUAL</td>
</tr>
<tr>
<td>MATH:VERTICAL:POSITION</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>MATH:VERTICAL:SCALE</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>MATH:VERTICAL:UNITS</td>
<td>&quot;V&quot;</td>
</tr>
<tr>
<td>MATHVAR:VAR1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>MATHVAR:VAR2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>MEASUREMENT:GATING</td>
<td>SCREEN</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:DELAY:DIRECTION</td>
<td>FORWARDS</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:DELAY:EDGE1</td>
<td>RISE</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:DELAY:EDGE2</td>
<td>RISE</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:SOURCE1</td>
<td>CH1</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:SOURCE2</td>
<td>CH2</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:TYPE</td>
<td>PERIOD</td>
</tr>
<tr>
<td>MEASUREMENT:INDICATORS:STATE</td>
<td>OFF</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS1:DELAY:DIRECTION</td>
<td>FORWARDS</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS1:DELAY:EDGE1</td>
<td>RISE</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS1:DELAY:EDGE2</td>
<td>RISE</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS1:SOURCE1</td>
<td>CH1</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS1:SOURCE2</td>
<td>CH2</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS1:STATE</td>
<td>0</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS1:TYPE</td>
<td>PERIOD</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS2:DELAY:DIRECTION</td>
<td>FORWARDS</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS2:DELAY:EDGE1</td>
<td>RISE</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS2:DELAY:EDGE2</td>
<td>RISE</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS2:SOURCE1</td>
<td>CH1</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS2:SOURCE2</td>
<td>CH2</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS2:STATE</td>
<td>0</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS2:TYPE</td>
<td>PERIOD</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS3:DELAY:DIRECTION</td>
<td>FORWARDS</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS3:DELAY:EDGE1</td>
<td>RISE</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS3:DELAY:EDGE2</td>
<td>RISE</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS3:SOURCE1</td>
<td>CH1</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS3:SOURCE2</td>
<td>CH2</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS3:STATE</td>
<td>0</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS3:TYPE</td>
<td>PERIOD</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS4:DELAY:DIRECTION</td>
<td>FORWARDS</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS4:DELAY:EDGE1</td>
<td>RISE</td>
</tr>
</tbody>
</table>
# Appendix C: Factory Defaults

## Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASUREMENT:MEAS4:DELAY:EDGE2</td>
<td>RISE</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS4:SOURCE1 CH1</td>
<td></td>
</tr>
<tr>
<td>MEASUREMENT:MEAS4:SOURCE2 CH2</td>
<td></td>
</tr>
<tr>
<td>MEASUREMENT:MEAS4:STATE</td>
<td>0</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS4:TYPE</td>
<td>PERIOD</td>
</tr>
<tr>
<td>MEASUREMENT:METHOD</td>
<td>AUTO</td>
</tr>
<tr>
<td>MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>MEASUREMENT:REFLEVEL:ABSOLUTE:LOW</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>MEASUREMENT:REFLEVEL:ABSOLUTE:MID1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>MEASUREMENT:REFLEVEL:ABSOLUTE:MID2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>MEASUREMENT:REFLEVEL:METHOD</td>
<td>PERCENT</td>
</tr>
<tr>
<td>MEASUREMENT:REFLEVEL:PERCENT:HIGH</td>
<td>90.0000</td>
</tr>
<tr>
<td>MEASUREMENT:REFLEVEL:PERCENT:LOW</td>
<td>10.0000</td>
</tr>
<tr>
<td>MEASUREMENT:REFLEVEL:PERCENT:MID1</td>
<td>50.0000</td>
</tr>
<tr>
<td>MEASUREMENT:REFLEVEL:PERCENT:MID2</td>
<td>50.0000</td>
</tr>
<tr>
<td>MEASUREMENT:STATISTICS:MODE</td>
<td>ALL</td>
</tr>
<tr>
<td>MEASUREMENT:STATISTICS:WEIGHTING</td>
<td>32</td>
</tr>
<tr>
<td>MESSAGE:BOX</td>
<td>167,67,167,83</td>
</tr>
<tr>
<td>MESSAGE:SHOW</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>MESSAGE:STATE</td>
<td>0</td>
</tr>
<tr>
<td>PICTBRIDGE:DATEPRINT</td>
<td>DEFLT</td>
</tr>
<tr>
<td>PICTBRIDGE:IDPRINT</td>
<td>OFF</td>
</tr>
<tr>
<td>PICTBRIDGE:IMAGESIZE</td>
<td>DE FLT</td>
</tr>
<tr>
<td>PICTBRIDGE:PAPERSIZE</td>
<td>DEFLT</td>
</tr>
<tr>
<td>PICTBRIDGE:PAPERTYPE</td>
<td>DEFLT</td>
</tr>
<tr>
<td>PICTBRIDGE:PRINTQUAL</td>
<td>DEFLT</td>
</tr>
<tr>
<td>REF1:HORIZONTAL:DELAY:TIME</td>
<td>16.0000E-6</td>
</tr>
<tr>
<td>REF1:</td>
<td>4.00000E-6</td>
</tr>
<tr>
<td>REF1:VERTICAL:POSITION</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>REF1:VERTICAL:SCALE</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>REF2:HORIZONTAL:DELAY:TIME</td>
<td>16.0000E-6</td>
</tr>
<tr>
<td>REF2:</td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>REF2:VERTICAL:POSITION</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>REF2:VERTICAL:SCALE</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>REF3:HORIZONTAL:DELAY:TIME</td>
<td>16.0000E-6</td>
</tr>
<tr>
<td>REF3:</td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>REF3:VERTICAL:POSITION</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>REF3:VERTICAL:SCALE</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>REF4:ORIZONTAL:DELAY:TIME</td>
<td>16.0000E-6</td>
</tr>
<tr>
<td>REF4:</td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>REF4:VERTICAL:POSITION</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>REF4:VERTICAL:SCALE</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>ROSC:SOURCE</td>
<td>INTERNAL</td>
</tr>
<tr>
<td>SAVE:ASSIGN:TYPE</td>
<td>WAVEFORM</td>
</tr>
<tr>
<td>SAVE:IMAGE:FILEFORMAT</td>
<td>PNG</td>
</tr>
<tr>
<td>SAVE:IMAGE:INKSAVER</td>
<td>1</td>
</tr>
<tr>
<td>SAVE:IMAGE:LAYOUT</td>
<td>PORTRAIT</td>
</tr>
<tr>
<td>SAVE:WAVEFORM</td>
<td>WFM</td>
</tr>
<tr>
<td>SAVE:WAVEFORM:GATING</td>
<td>NONE</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:STATE</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
</table>
Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
</table>
## Appendix C: Factory Defaults

### Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------------</td>
</tr>
</tbody>
</table>
### Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:BUS:SOURCE</td>
<td>B1</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE</td>
<td>RISE</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:EDGE:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LEVEL</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LEVEL:MATH</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:FUNCTION AND</td>
<td></td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH1</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D0</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D1</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D2</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF1</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D0</td>
<td>X</td>
</tr>
</tbody>
</table>
Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:WHEN true</td>
<td></td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:MATH</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:MATH</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF4</td>
<td>0.0E+0</td>
</tr>
</tbody>
</table>
## Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:POLARITY</td>
<td>POSITIVE</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:WHEN</td>
<td>LESS THAN</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY</td>
<td>POSITIVE</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEN</td>
<td>OCCURS</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:MATH</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY</td>
<td>STAYSHIGH</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:TRANSITION:POLARITY</td>
<td>POSITIVE</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEN</td>
<td>SLOWER</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:TYPE</td>
<td>EDGE</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH1</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH2</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH3</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH4</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:MATH</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF1</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF2</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF3</td>
<td>800.0000E-3</td>
</tr>
</tbody>
</table>
### Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF4</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGGER:A:BUS:B&lt;x&gt;:CAN:DATA:OFFSET</td>
<td>0</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:AUDIO:CONDITION</td>
<td>SOF</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:AUDIO:DATA:OFFSET</td>
<td>0</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:CAN:DATA:SIZE</td>
<td>1</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:CAN:DATA:START</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:FLEXRAY:CONDITION</td>
<td>SOF</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:FLEXRAY:DATA:SIZE</td>
<td>1</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:FLEXRAY:ERRTYPE</td>
<td>CRCHEADER</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:FLEXRAY:FRAMETYPE</td>
<td>NORMAL</td>
</tr>
</tbody>
</table>
Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE</td>
<td>&quot;XXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:I2C:CONDITION</td>
<td>START</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:I2C:DATA:SIZE</td>
<td>1</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:I2C:DATA:START</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:I2C:DATA:VALUE</td>
<td>&quot;XXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:LIN:CONDITION</td>
<td>SYNCFIELD</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:LIN:DATA:SIZE</td>
<td>1</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:LIN:ERRTYPE</td>
<td>SYNC</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE</td>
<td>&quot;XXXXXX&quot;</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:PARALLEL:VALUE</td>
<td>&quot;XXXXXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:SPI:CONDITION</td>
<td>SS</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:SPI:DATA:SIZE</td>
<td>1</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:SPI:DATA:START</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:USB:CONDITION</td>
<td>SYNC</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:USB:DATA:SIZE</td>
<td>1</td>
</tr>
</tbody>
</table>
### Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGGER:A:BUS:B1:USB:ERRTYPE</td>
<td>PID</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:USB:HANDSHAKETYPE</td>
<td>ANY</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:USB:QUALIFIER</td>
<td>EQUAL</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:USB:SOFRAMENUMBER</td>
<td>&quot;XXXXXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:USB:SPECIALTYPE</td>
<td>ANY</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B1:USB:TOKENTYPE</td>
<td>ANY</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B2:CAN:DATA:START</td>
<td>0.0E+0</td>
</tr>
</tbody>
</table>
### Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGGER:A:BUS:B2:I2C:DATA:START</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:BUS:B2:SPI:DATA:START</td>
<td>0.0E+0</td>
</tr>
</tbody>
</table>
Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGGER:A:EDGE:COUPLING</td>
<td>DC</td>
</tr>
<tr>
<td>TRIGGER:A:EDGE:SLOPE</td>
<td>RISE</td>
</tr>
<tr>
<td>TRIGGER:A:EDGE:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>TRIGGER:A:HOLDOFF:TIME</td>
<td>20.0000E-9</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D0</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D1</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D10</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D11</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D12</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D13</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D14</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D15</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D2</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D3</td>
<td>1.4000</td>
</tr>
</tbody>
</table>
Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGGER:A:LEVEL:D4</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D5</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D6</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D7</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D8</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LEVEL:D9</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:CLASS</td>
<td>LOGIC</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:FUNCTION</td>
<td>AND</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:CH1</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:CH2</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:CH3</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:CH4</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:CLOCK:EDGE</td>
<td>RISE</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:CLOCK:SOURCE</td>
<td>NONE</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D0</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D1</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D10</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D11</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D12</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D13</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D14</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D15</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D2</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D3</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D4</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D5</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D6</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D7</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D8</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:INPUT:D9</td>
<td>X</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:PATTERN:DELTATIME</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:PATTERN:WHEN</td>
<td>TRUE</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:CH3</td>
<td>0.0E+0</td>
</tr>
</tbody>
</table>
Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D0</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D1</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D10</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D11</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D12</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D13</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D14</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D15</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D2</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D3</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D4</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D5</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D6</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D7</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D8</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOGIC:THRESHOLD:D9</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D0</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D1</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D10</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D11</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D12</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D13</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D14</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D15</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D2</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D3</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D4</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D5</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D6</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D7</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D8</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:LOWERTHRESHOLD:D9</td>
<td>1.4000</td>
</tr>
</tbody>
</table>
### Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGGER:A:MODE</td>
<td>AUTO</td>
</tr>
<tr>
<td>TRIGGER:A:PULSE:CLASS</td>
<td>WIDTH</td>
</tr>
<tr>
<td>TRIGGER:A:PULSEWIDTH:HIGHLIMIT</td>
<td>12.0000E-9</td>
</tr>
<tr>
<td>TRIGGER:A:PULSEWIDTH:LOWLIMIT</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIGGER:A:PULSEWIDTH:POLARITY</td>
<td>POSITIVE</td>
</tr>
<tr>
<td>TRIGGER:A:PULSEWIDTH:WHEN</td>
<td>LESSTHAN</td>
</tr>
<tr>
<td>TRIGGER:A:PULSEWIDTH:WIDTH</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIGGER:A:RUNT:POLARITY</td>
<td>POSITIVE</td>
</tr>
<tr>
<td>TRIGGER:A:RUNT:WHEN</td>
<td>OCCURS</td>
</tr>
<tr>
<td>TRIGGER:A:RUNT:WIDTH</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:CLOCK:EDGE</td>
<td>RISE</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:CLOCK:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:CLOCK:THRESHOLD</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:DATA:SOURCE</td>
<td>NONE</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:DATA:THRESHOLD</td>
<td>9.91E+37</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:HOLDTIME</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:SETTIME</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D0</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D1</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D10</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D11</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D12</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D13</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D14</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D15</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D2</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D3</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D4</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D5</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D6</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D7</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D8</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:A:SETHOLD:THRESHOLD:D9</td>
<td>1.4000</td>
</tr>
</tbody>
</table>
### Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGGER:A:TIMEOUT:POLARITY</td>
<td>STAYSHIGH</td>
</tr>
<tr>
<td>TRIGGER:A:TIMEOUT:TIME</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIGGER:A:TRANSITION:DELTATIME</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIGGER:A:TRANSITION:POLARITY</td>
<td>POSITIVE</td>
</tr>
<tr>
<td>TRIGGER:A:TRANSITION:WHEN</td>
<td>SLOWER</td>
</tr>
<tr>
<td>TRIGGER:A:TYPE</td>
<td>EDGE</td>
</tr>
<tr>
<td>TRIGGER:A:UPPERTHRESHOLD:CH1</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>TRIGGER:A:UPPERTHRESHOLD:CH2</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>TRIGGER:A:UPPERTHRESHOLD:CH3</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>TRIGGER:A:UPPERTHRESHOLD:CH4</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>TRIGGER:B:BY</td>
<td>TIME</td>
</tr>
<tr>
<td>TRIGGER:B:EDGE:COUPLING</td>
<td>DC</td>
</tr>
<tr>
<td>TRIGGER:B:EDGE:SLOPE</td>
<td>RISE</td>
</tr>
<tr>
<td>TRIGGER:B:EDGE:SOURCE</td>
<td>CH1</td>
</tr>
<tr>
<td>TRIGGER:B:EVENTS:COUNT</td>
<td>1</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D0</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D1</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D10</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D11</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D12</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D13</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D14</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D15</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D2</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D3</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D4</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D5</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D6</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D7</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D8</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LEVEL:D9</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWERTHRESHOLD:CH1</td>
<td>0.0E+0</td>
</tr>
</tbody>
</table>
### Table C-1: Default Values (cont.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D0</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D1</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D10</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D11</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D12</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D13</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D14</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D15</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D2</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D3</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D4</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D5</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D6</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D7</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D8</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:LOWER_THRESHOLD:D9</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIGGER:B:STATE</td>
<td>0</td>
</tr>
<tr>
<td>TRIGGER:B:TIME</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIGGER:B:TYPE</td>
<td>EDGE</td>
</tr>
<tr>
<td>VERBOSE</td>
<td>1</td>
</tr>
<tr>
<td>VIDPIC:AUTOCONTRAST</td>
<td>1</td>
</tr>
<tr>
<td>VIDPIC:AUTOCONTRAST:UPDATERATE</td>
<td>20</td>
</tr>
<tr>
<td>ZOOM:MODE</td>
<td>0</td>
</tr>
<tr>
<td>ZOOM:ZOOM1:POSITION</td>
<td>50.0000</td>
</tr>
<tr>
<td>ZOOM:ZOOM1:SCALE</td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>ZOOM:ZOOM1:STATE</td>
<td>0</td>
</tr>
</tbody>
</table>
Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples

Here are several examples of command sequences using the WFMOutpre? and CURVe? queries to transfer data from the oscilloscope to a PC, with different data sources, including Analog, Digital waveforms. Each command sequence example is followed by an explanation of the returned WFMOutpre? results. The WFMOutpre? values act as settings that apply to the CURVe query data being transferred.

Here are several examples of command sequences using the WFMOutpre? and CURVe? queries to transfer data from the oscilloscope to a PC, with different data sources, including Analog, Digital, and RF frequency domain waveforms. Each command sequence example is followed by an explanation of the returned WFMOutpre? results. The WFMOutpre? values act as settings that apply to the CURVe query data being transferred.

NOTE. In order to guarantee that the waveform data returned from CURVe? queries of multiple waveforms are correlated to the same acquisition, you should use single sequence acquisition mode to acquire the waveform data from a single acquisition. Single sequence acquisition mode is enabled using ACQuire:STOPAfter SEQuence.

Example 1: Analog Waveform (Channels 1–4)

Goal: Transfer 10,000 points of analog channel waveform data from the oscilloscope to a PC.

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DATa:SOUrce CH1</td>
<td></td>
</tr>
<tr>
<td>:DATa:START 1</td>
<td></td>
</tr>
<tr>
<td>:DATa:STOP 10000</td>
<td></td>
</tr>
<tr>
<td>:WFMOutpre:ENCdg BINARY</td>
<td></td>
</tr>
<tr>
<td>:WFMOutpre:BYT_Nr 1</td>
<td></td>
</tr>
<tr>
<td>:HEAder 1</td>
<td></td>
</tr>
</tbody>
</table>
Appendix D: Waveform Transfer (WFMOoutpre and CURVe Query) Examples

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>:WFMOoutpre?</td>
<td>Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the oscilloscope to a PC or other device (see next table for explanations):</td>
</tr>
<tr>
<td></td>
<td>:WFMOoutpre:BYT_NR 1;BIT_NR 8;ENCdg BINARY;BN_FMT RI;BYT_OR MSB;WFID &quot;Ch1, DC coupling, 100.0mV/div, 4.000us/div, 10000 points, Sample mode&quot;;NR_PT 10000;PT_FMT Y;PT_ORDER LINEAR;XUNIT &quot;s&quot;;XINCR 4.0000E-9;XZERO -20.0000E-6;PT_OFF 0;YUNIT &quot;V&quot;;YMULT 4.0000E-3;YOFF 0.0E+0;YZERO 0.0E+0</td>
</tr>
<tr>
<td>:CURVe?</td>
<td>Returns 10,000 data points:</td>
</tr>
<tr>
<td></td>
<td>:CURVe #510000&lt;10,000 binary bytes of waveform data.&gt;</td>
</tr>
</tbody>
</table>

**NOTE.** You can also use the WAVFrm? query, which concatenates the WFMOoutpre? and CURVe? queries.

<table>
<thead>
<tr>
<th>WFMOoutpre? Query results</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYT_NR 1</td>
<td>This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOoutpre:BYT_Nr command. Note that changing this value automatically changes the BIT_NR value accordingly. (This value can also be set using the DATA:WIDTH command.)</td>
</tr>
<tr>
<td>BIT_NR 8</td>
<td>This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOoutpre:BIT_Nr command. Note that changing this value automatically changes the BYT_NR value accordingly.</td>
</tr>
<tr>
<td>ENCDG BINARY</td>
<td>This value specifies the encoding of the waveform data. To change this value (the other possibility is ASCii), use the WFMOoutpre:ENCdg command. (This value can also be set using the DATA:ENCdg command, which provides the ability to set the WFMOoutpre:ENCdg, WFMOoutpre:BN_FMT, and WFMOoutpre:BYT_OR values using a single command.)</td>
</tr>
<tr>
<td>BN_FMT RI</td>
<td>This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the WFMOoutpre:BN_FMT command. Note: this field is not applicable for ASCii encoding.</td>
</tr>
<tr>
<td>BYT_OR MSB</td>
<td>This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first). To change this value to LSB, use the WFMOoutpre:BYT_OR command. Note: this field is not applicable for ASCii encoding.</td>
</tr>
<tr>
<td>WFID &quot;Ch1, DC coupling, 100.0mV/div, 4.000us/div, 10000 points, Sample mode&quot;</td>
<td>This quoted string represents information about the source waveform that would be returned by a WFMOoutpre:WFID? query.</td>
</tr>
<tr>
<td>NR_PT 10000</td>
<td>This value indicates the number of data points in the waveform record to be transferred using the CURVe? query. (If you would like to determine only this value, use the WFMOoutpre:NR_PT? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATA:START and DATA:STOP commands.</td>
</tr>
</tbody>
</table>
Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples

<table>
<thead>
<tr>
<th>WFMOutpre? Query results</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PT_FMT Y</td>
<td>This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only – the returned values can be Y for YT format or ENV for envelope format (min/max pairs). (If you would like to determine only this value, use the WFMOutpre:PT_Fmt? query.)</td>
</tr>
<tr>
<td>PT_ORDER LINEAR</td>
<td>This value is always LINEAR.</td>
</tr>
<tr>
<td>XUNIT &quot;s&quot;</td>
<td>This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. Typically, this value is &quot;s&quot;, representing seconds. When using the math waveform as a source, the value can be &quot;s&quot; or &quot;Hz&quot;. This is query only. (If you would like to determine only this value, use the WFMOutpre:XUNit? query.)</td>
</tr>
<tr>
<td>XINCR 4.0000E-9</td>
<td>This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOutpre:XINcr? query.)</td>
</tr>
<tr>
<td>XZERO -20.0000E-6</td>
<td>This value indicates the time, in seconds, or frequency, in hertz, of the first data point in the waveform record. This time or frequency is relative to the time of the trigger, which is always 0. So, this XZERO time or frequency can be negative. This is query only. (If you would like to determine only this value, use the WFMOutpre:XZEro? query.)</td>
</tr>
<tr>
<td>PT_OFF 0</td>
<td>This is a query provided only for compatibility with performance oscilloscopes. The returned value is always 0. (If you would like to determine only this value, use the WFMOutpre:PT_Off? query.)</td>
</tr>
<tr>
<td>YUNIT &quot;V&quot;</td>
<td>This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, volts. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNit? query.)</td>
</tr>
<tr>
<td>YMULT 4.0000E-3</td>
<td>This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the WFMOutpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMUlt? query.)</td>
</tr>
<tr>
<td>YOFF 0.0E+0</td>
<td>This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOutpre:YOFf? query.)</td>
</tr>
<tr>
<td>YZERO 0.0E+0</td>
<td>This value indicates the vertical offset of the source waveform in units specified by the WFMOutpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YZEro? query.)</td>
</tr>
</tbody>
</table>

Example 2: Digital Waveform

Goal: Transfer 25 points of digital channel waveform data from the oscilloscope to a PC.

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DATa:SOURCe CH1_D5</td>
<td></td>
</tr>
<tr>
<td>:DATa:START 1</td>
<td></td>
</tr>
<tr>
<td>:DATa:STOP 25</td>
<td></td>
</tr>
<tr>
<td>:WFMOutpre:ENCdg ASCii</td>
<td></td>
</tr>
<tr>
<td>:WFMOutpre:BYT_Nr 1</td>
<td></td>
</tr>
</tbody>
</table>
### Command | Comment
--- | ---
:HEADer 1 | Returns the following values. Each value represents the current settings that a CURve? query will use to format the data that will be transferred from the oscilloscope to a PC or other device (see next table for explanations):
:WFMOutpre? | :WFMOUTPRE:BYT_NR 1;BIT_NR 8;ENCdg ASCII;BN_FMT RI;BYT_Or MSB;WFID "D5, unknown coupling, 100.0us/div, 10000 points, Digital mode";NR_PT 25;PT_FMT Y;PT_ORDER LINEAR;XUNIT "s";XINCR 100.0000E-9;XZERO -500.0000E-6;PT_OFF 0;YUNIT "State";YMULT 1.0000;YOFF 0.0E+0;YZERO 0.0E+0
:CURve? | Returns the following values. Each value represents a data point:
:CURve 0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0

**NOTE.** You can also use the WAVFrm? query, which concatenates the WFMOutpre? and CURve? queries.

### WFMOutpre? Query results

<table>
<thead>
<tr>
<th>WFMOutpre?</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFMOUTPRE:BYT_NR 1</td>
<td>This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOutpre:BYT_Nr command. Note that changing this value automatically changes the BIT_NR value accordingly. (This value can also be set using the DATA:WIDTH command.)</td>
</tr>
<tr>
<td>BIT_NR 8</td>
<td>This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOutpre:BIT_Nr command. Note that changing this value automatically changes the BYT_NR value accordingly.</td>
</tr>
<tr>
<td>ENCDG ASCII</td>
<td>This value specifies the encoding of the waveform data. To change this value (the other possibility is BINARY), use the WFMOutpre:ENCdg command. (This value can also be set using the DATA:ENCdg command, which provides the ability to set the WFMOutpre:ENCdg, WFMOutpre:BN_FMT, and WFMOutpre:BYT_Or values using a single command.)</td>
</tr>
<tr>
<td>BN_FMT RI</td>
<td>This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the WFMOutpre:BN_FMT command. Note: this field is not applicable for ASCII encoding.</td>
</tr>
<tr>
<td>BYT_Or MSB</td>
<td>This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOutpre:BYT_Or command. Note: this field is not applicable for ASCII encoding.</td>
</tr>
<tr>
<td>WFID &quot;D5, unknown coupling, 100.0us/div, 10000 points, Digital mode&quot;</td>
<td>This quoted string represents information about the source waveform that would be returned by a WFMOutpre:WFId? query. It cannot be changed.</td>
</tr>
</tbody>
</table>
Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples

WFMOutpre? Query results | Explanation
--- | ---
NR_PT 25 | This value indicates the number of data points in the waveform record to be transferred using the CURVE? query. (If you would like to determine only this value, use the WFMOutpre:NR_PT? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATa:START and DATa:STOP commands.

PT_FMT Y | This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only. (If you would like to determine only this value, use the WFMOutpre:PT_FMT? query.)

PT_ORDER LINEAR | This value is always LINEAR.

XUNIT "s" | This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. Typically, this value is "s", representing seconds. When using the math waveform as a source, the value can be "s" or "Hz". This is query only. (If you would like to determine only this value, use the WFMOutpre:XUNIT? query.)

XINCR 100.0000E-9 | This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOutpre:XINCR? query.)

XZERO -500.0000E-6 | This value indicates the time, in seconds, or frequency, in hertz, of the first data point in the waveform record. This time or frequency is relative to the time of the trigger, which is always 0. So, this XZERO time or frequency can be negative. This is query only. (If you would like to determine only this value, use the WFMOutpre:XZERO? query.)

PT_OFF 0 | This is a query provided only for compatibility with other Tektronix oscilloscopes. The returned value is always 0. (If you would like to determine only this value, use the WFMOutpre:PT_OFF? query.)

YUNIT "State" | This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, State. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNIT? query.)

YMULT 1.0000 | This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the WFMOutpre:YUNIT command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMULT? query.)

YOFF 0.0E+0 | This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOutpre:YOFF? query.)

YZERO 0.0E+0 | This value indicates the vertical offset of the source waveform in units specified by the WFMOutpre:YUNIT command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YZERO? query.)

Example 3: Digital with 4 Bytes Per Point and Zoom Off

Goal: Transfer 25 points of Digital data from the oscilloscope to a PC using 4 bytes per point and Zoom off.
Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DATa:SOURce CH1_D1</td>
<td></td>
</tr>
<tr>
<td>:ACQuire:MAGnivu 0</td>
<td></td>
</tr>
<tr>
<td>:DATa:START 1</td>
<td></td>
</tr>
<tr>
<td>:DATa:STOP 25</td>
<td></td>
</tr>
<tr>
<td>:WFMOutpre:ENCdg ASCII</td>
<td></td>
</tr>
<tr>
<td>:WFMOutpre:BYT_Nr 4</td>
<td></td>
</tr>
<tr>
<td>:HEADer 1</td>
<td></td>
</tr>
<tr>
<td>:VERBose 1</td>
<td></td>
</tr>
</tbody>
</table>
| :WFMOutpre? | Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the oscilloscope to a PC or other device (see next table for explanations): 
:WFMOUTPRE:BYT_NR 4;BIT_NR 32;ENCDG ASCII;BN_FMT RI;BYT_OR MSB;WFID "Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode";NR_PT 25;PT_FMT Y;PT_ORDER LINEAR;XUNIT "s";XINCR 100.0000E-9;XZERO -500.0000E-6;PT_OFF 0;YUNIT "State";YMULT 1.0000;YOFF 0.0E+0;YZERO 0.0E+0 |
| :CURVe? | Returns the following values. Each value represents a data point: 
:CURVe

**NOTE.** The returned hexadecimal data values are formatted without leading zeroes. For example, a 4-byte digital value of FB386 should be interpreted as 000FB386.

**NOTE.** You can also use the WAVFrm? query, which concatenates the WFMOutpre? and CURVe? queries.

<table>
<thead>
<tr>
<th>WFMOutpre? Query results</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFMOUTPRE:BYT_NR 4</td>
<td>This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOutpre:BYT_Nr command. Note that changing this value automatically changes the BIT_NR value accordingly. (This value can also be set using the DATA:WIDTH command.)</td>
</tr>
<tr>
<td>BIT_NR 32</td>
<td>This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOutpre:BIT_Nr command. Note that changing this value automatically changes the BYT_NR value accordingly.</td>
</tr>
</tbody>
</table>
### Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples

<table>
<thead>
<tr>
<th>WFMOutpre? Query Results</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENCDG ASCII</td>
<td>This value specifies the encoding of the waveform data. To change this value (the other possibility is BINARY), use the WFMOutpre:ENCdg command. (This value can also be set using the DATa:ENCdg command, which provides the ability to set the WFMOutpre:ENCdg, WFMOutpre:BN_FMT, and WFMOutpre:BYT_OR values using a single command.)</td>
</tr>
<tr>
<td>BN_FMT RI</td>
<td>This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the WFMOutpre:BN_FMT command. Note: this field is not applicable for ASCII encoding.</td>
</tr>
<tr>
<td>BYT_OR MSB</td>
<td>This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOutpre:BYT_OR command. Note: this field is not applicable for ASCII encoding.</td>
</tr>
<tr>
<td>WFID &quot;Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode&quot;</td>
<td>This quoted string represents information about the source waveform that would be returned by a WFMOutpre:WFID? query. It cannot be changed.</td>
</tr>
<tr>
<td>NR_PT 25</td>
<td>This value indicates the number of data points in the waveform record to be transferred using the CURVe? query. (If you would like to determine only this value, use the WFMOutpre:NR_PT? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATA:START and DATA:STOP commands.</td>
</tr>
<tr>
<td>PT_FMT Y</td>
<td>This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only – the returned values can be Y for YT format or ENV for envelope format (min/max pairs). (If you would like to determine only this value, use the WFMOutpre:PT_FMT? query.)</td>
</tr>
<tr>
<td>PT_ORDER LINEAR</td>
<td>This value is always LINEar.</td>
</tr>
<tr>
<td>XUNIT &quot;s&quot;</td>
<td>This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. For live channels, this value is &quot;s&quot;, representing seconds. When using the math waveform as a source, the value can be &quot;s&quot; or &quot;Hz&quot;. This is query only. (If you would like to determine only this value, use the WFMOutpre:XUNIT? query.)</td>
</tr>
<tr>
<td>XINCR 100.0000E-9</td>
<td>This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOutpre:XINCR? query.)</td>
</tr>
<tr>
<td>XZERO -500.0000E-6</td>
<td>This value indicates the time, in seconds, or frequency, in hertz, of the first data point in the waveform record. This time or frequency is relative to the time of the trigger, which is always 0. So, this XZERO time or frequency can be negative. This is query only. (If you would like to determine only this value, use the WFMOutpre:XZERO? query.)</td>
</tr>
<tr>
<td>PT_OFF 0</td>
<td>This is a query provided only for compatibility with performance oscilloscopes. The returned value is always 0. (If you would like to determine only this value, use the WFMOutpre:PT_OFF? query.)</td>
</tr>
<tr>
<td>YUNIT &quot;State&quot;</td>
<td>This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, State. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNIT? query.)</td>
</tr>
</tbody>
</table>
### Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples

#### WFMOutpre? Query results

<table>
<thead>
<tr>
<th>Query</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>YMULT 1.0000</td>
<td>This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the WFMOutpre:YUNIT command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMULT? query.)</td>
</tr>
<tr>
<td>YOFF 0.0E+0</td>
<td>This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOutpre:YOFF? query.)</td>
</tr>
<tr>
<td>YZERO 0.0E+0</td>
<td>This value indicates the vertical offset of the source waveform in units specified by the WFMOutpre:YUNIT command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YZERO? query.)</td>
</tr>
</tbody>
</table>

#### Example 4: Digital with 8 Bytes Per Point and Zoom Off

**Goal:** Transfer 25 points of Digital data from the oscilloscope to a PC using 8 bytes per point and Zoom off.

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DATa:SOURce CH1_D1</td>
<td></td>
</tr>
<tr>
<td>:ACQuire:MAGnivu 0</td>
<td></td>
</tr>
<tr>
<td>:DATa:START 1</td>
<td></td>
</tr>
<tr>
<td>:DATa:STOP 25</td>
<td></td>
</tr>
<tr>
<td>:WFMOutpre:ENCdg ASCII</td>
<td></td>
</tr>
<tr>
<td>:WFMOutpre:BYT_NR 8</td>
<td></td>
</tr>
<tr>
<td>:HEADer 1</td>
<td></td>
</tr>
<tr>
<td>:VERBose 1</td>
<td></td>
</tr>
<tr>
<td>:WFMOutpre?</td>
<td>Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the oscilloscope to a PC or other device (see next table for explanations):</td>
</tr>
<tr>
<td>:WFMOUTPRE:BYT_NR 8;BIT_NR 64;ENCdg ASCII;BN_FMT RI;BYT.OR MSB;WFID &quot;Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode&quot;;NR_PT 25;PT_FMT Y;PT_ORDER LINEAR;XUNIT &quot;s&quot;;XINCR 100.0000E-9;YZERO -500.0000E-6;PT_OFF 0;YUNIT &quot;State&quot;;YMULT 1.0000;YOFF 0.0E+0;YZERO 0.0E+0</td>
<td></td>
</tr>
<tr>
<td>:CURVe?</td>
<td>Returns the following values. Each value represents a data point:</td>
</tr>
<tr>
<td>:CURVe 80000FB386,E000FB386,80000FB3E6,80000FB3E6,80000FB3E6,80000FB3E6,C8000FB3A6,8C000FB3A6,8C000FB3A6,84000FB3AE,CC000FB3A6,8C000FB3E6,8C000FB3E6,84000FB3E6,80000FB3E6,80000FB3E6,80000FB3E6,A000FB3C6,80000FB3C6,80000FB3C6,80000FB3C6,80000FB3C6,88000FB3C6,8C000FB3C6,8C000FB3C6,84000FB3CE</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** The returned hexadecimal data values are formatted without leading zeroes.
**NOTE.** You can also use the `WAVFrm?` query, which concatenates the `WFMOutpre?` and `CURVe?` queries.

<table>
<thead>
<tr>
<th>WFMOutpre? Query results</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>WFMOutpre:BYT_NR 8</code></td>
<td>This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the <code>WFMOutpre:BYT_Nr</code> command. Note that changing this value automatically changes the <code>BIT_NR</code> value accordingly. (This value can also be set using the <code>DATa:WIDth</code> command.)</td>
</tr>
<tr>
<td><code>BIT_NR 64</code></td>
<td>This value specifies the number of bits per data point in the waveform data. To change this value, use the <code>WFMOutpre:BIT_Nr</code> command. Note that changing this value automatically changes the <code>BYT_NR</code> value accordingly.</td>
</tr>
<tr>
<td><code>ENCDG ASCII</code></td>
<td>This value specifies the encoding of the waveform data. To change this value (the other possibility is BINARY), use the <code>WFMOutpre:ENCdg</code> command. (This value can also be set using the <code>DATa:ENCdg</code> command, which provides the ability to set the <code>WFMOutpre:ENCdg</code>, <code>WFMOutpre:BN_FMT</code>, and <code>WFMOutpre:BYT_OR</code> values using a single command.)</td>
</tr>
<tr>
<td><code>BN_FMT RI</code></td>
<td>This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the <code>WFMOutpre:BN_FMT</code> command. Note: this field is not applicable for ASCII encoding.</td>
</tr>
<tr>
<td><code>BYT_OR MSB</code></td>
<td>This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the <code>WFMOutpre:BYT_OR</code> command. Note: this field is not applicable for ASCII encoding.</td>
</tr>
<tr>
<td><code>WFID &quot;Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode&quot;</code></td>
<td>This quoted string represents information about the source waveform that would be returned by a <code>WFMOutpre:WFId?</code> query. It cannot be changed.</td>
</tr>
<tr>
<td><code>NR_PT 25</code></td>
<td>This value indicates the number of data points in the waveform record to be transferred using the <code>CURVe?</code> query. (If you would like to determine only this value, use the <code>WFMOutpre:NR_PT?</code> query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the <code>DATa:STARt</code> and <code>DATa:STOP</code> commands.</td>
</tr>
<tr>
<td><code>PT_FMT Y</code></td>
<td>This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only – the returned values can be Y for YT format or ENV for envelope format (min/max pairs). (If you would like to determine only this value, use the <code>WFMOutpre:PT_FMT?</code> query.)</td>
</tr>
<tr>
<td><code>PT_ORDER LINEAR</code></td>
<td>This value is always LINEar.</td>
</tr>
<tr>
<td><code>XUNIT &quot;s&quot;</code></td>
<td>This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. Typically, this value is &quot;s&quot;, representing seconds. When using the math waveform as a source, the value can be &quot;s&quot; or &quot;Hz&quot;. This is query only. (If you would like to determine only this value, use the <code>WFMOutpre:XUNit?</code> query.)</td>
</tr>
<tr>
<td><code>XINCR 100.0000E-9</code></td>
<td>This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the <code>WFMOutpre:XINcr?</code> query.)</td>
</tr>
</tbody>
</table>
Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples

<table>
<thead>
<tr>
<th>WFMOutpre? Query results</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>XZER0 -500.0000E-6</td>
<td>This value indicates the time, in seconds, or frequency, in hertz, of the first data point in the waveform record. This time or frequency is relative to the time of the trigger, which is always 0. So, this XZER0 time or frequency can be negative. This is query only. (If you would like to determine only this value, use the WFMOutpre:XZER0? query.)</td>
</tr>
<tr>
<td>PT_OFF 0</td>
<td>This is a query provided only for compatibility with other Tektronix oscilloscopes. The returned value is always 0. (If you would like to determine only this value, use the WFMOutpre:PT_Off? query.)</td>
</tr>
<tr>
<td>YUNIT &quot;State&quot;</td>
<td>This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, State. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNIT? query.)</td>
</tr>
<tr>
<td>YMULT 1.0000</td>
<td>This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the WFMOutpre:YUNIT command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMULT? query.)</td>
</tr>
<tr>
<td>YOFF 0.0E+0</td>
<td>This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOutpre:YOFF? query.)</td>
</tr>
<tr>
<td>YZERO 0.0E+0</td>
<td>This value indicates the vertical offset of the source waveform in units specified by the WFMOutpre:YUNIT command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YZERO? query.)</td>
</tr>
</tbody>
</table>

**Example 5: Digital with 4 Bytes Per Point and Zoom On**

**Goal:** Transfer 25 points of Digital data from the oscilloscope to a PC using 4 bytes per point and Zoom on.

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DATA:SOURCE CH1_D1</td>
<td></td>
</tr>
<tr>
<td>:ACQuire:MAGnivu 1</td>
<td></td>
</tr>
<tr>
<td>:DATA:START 1</td>
<td></td>
</tr>
<tr>
<td>:DATA:STOP 25</td>
<td></td>
</tr>
<tr>
<td>:WFMOOutpre:ENCdg ASCii</td>
<td></td>
</tr>
<tr>
<td>:WFMOOutpre:BYT_Nr 4</td>
<td></td>
</tr>
<tr>
<td>:HEADer 1</td>
<td></td>
</tr>
<tr>
<td>:VERBose 1</td>
<td></td>
</tr>
</tbody>
</table>
### Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFMOutpre?</td>
<td>Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the oscilloscope to a PC or other device (see next table for explanations):</td>
</tr>
<tr>
<td></td>
<td>:WFMOutpre:BYT_NR 4;BIT_NR 32;ENCDG ASCII;BN_FMT RI;BYT_OR MSB;WID &quot;Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode&quot;;NR_PT 25;PT_FMT Y;PT_ORDER LINEAR;XUNIT &quot;s&quot;;XINC 1.2121E-9;XZERO -6.0606E-6;PT_OFF 0;YUNIT &quot;State&quot;;YMULT 1.0000;YOFF 0.0E+0</td>
</tr>
<tr>
<td>CURVe?</td>
<td>Returns the following values. Each value represents a data point:</td>
</tr>
<tr>
<td></td>
<td>:CURVe FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6,FB6E6</td>
</tr>
</tbody>
</table>

**NOTE.** The returned hexadecimal data values are formatted without leading zeroes.

**NOTE.** You can also use the WAVFrm? query, which concatenates the WFMOutpre? and CURVe? queries.

### WFMOutpre? Query results

<table>
<thead>
<tr>
<th>WFMOutpre?</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYT_NR 4</td>
<td>This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOutpre:BYT_Nr command. Note that changing this value automatically changes the BIT_NR value accordingly. (This value can also be set using the DATA:WIDth command.)</td>
</tr>
<tr>
<td>BIT_NR 32</td>
<td>This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOutpre:BIT_Nr command. Note that changing this value automatically changes the BYT_NR value accordingly.</td>
</tr>
<tr>
<td>ENCDG ASCII</td>
<td>This value specifies the encoding of the waveform data. To change this value (the other possibility is BINARY), use the WFMOutpre:ENCdg command. (This value can also be set using the DATA:ENCdg command, which provides the ability to set the WFMOutpre:ENCdg, WFMOutpre:BN_FMT, and WFMOutpre:BYT_OR values using a single command.)</td>
</tr>
<tr>
<td>BN_FMT RI</td>
<td>This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the WFMOutpre:BN_FMT command. Note: this field is not applicable for ASCII encoding.</td>
</tr>
<tr>
<td>BYT_OR MSB</td>
<td>This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOutpre:BYT_OR command. Note: this field is not applicable for ASCII encoding.</td>
</tr>
</tbody>
</table>
## Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples

<table>
<thead>
<tr>
<th>WFMOutpre? Query results</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>WID &quot;digital, unknown coupling, 100.0us/div, 10000 points, Digital mode&quot;</td>
<td>This quoted string represents information about the source waveform that would be returned by a WFMOutpre:WFID? query. It cannot be changed.</td>
</tr>
<tr>
<td>NR_PT 25</td>
<td>This value indicates the number of data points in the waveform record to be transferred using the CURVE? query. (If you would like to determine only this value, use the WFMOutpre:NR_PT? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATa:START and DATa:STOP commands.</td>
</tr>
<tr>
<td>PT_FMT Y</td>
<td>This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only – the returned values can be Y for YT format or ENV for envelope format (min/max pairs). (If you would like to determine only this value, use the WFMOutpre:PT_FMT? query.)</td>
</tr>
<tr>
<td>PT_ORDER LINEAR</td>
<td>This value is always LINEar.</td>
</tr>
<tr>
<td>XUNIT &quot;s&quot;</td>
<td>This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. Typically, this value is &quot;s&quot;, representing seconds. When using the math waveform as a source, the value can be &quot;s&quot; or &quot;Hz&quot;. This is query only. (If you would like to determine only this value, use the WFMOutpre:XUNIT? query.)</td>
</tr>
<tr>
<td>XINCR 1.2121E-9</td>
<td>This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOutpre:XINCR? query.)</td>
</tr>
<tr>
<td>XZERO -6.0606E-6</td>
<td>This value indicates the time, in seconds, or frequency, in hertz, of the first data point in the waveform record. This time or frequency is relative to the time of the trigger, which is always 0. So, this XZeRO time or frequency can be negative. This is query only. (If you would like to determine only this value, use the WFMOutpre:XZERO? query.)</td>
</tr>
<tr>
<td>PT_OFF 0</td>
<td>This is a query provided only for compatibility with performance oscilloscopes. The returned value is always 0. (If you would like to determine only this value, use the WFMOutpre:PT_OFF? query.)</td>
</tr>
<tr>
<td>YUNIT &quot;State&quot;</td>
<td>This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, State. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNIT? query.)</td>
</tr>
<tr>
<td>YMULT 1.0000</td>
<td>This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the WFMOutpre:YMULT command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMULT? query.)</td>
</tr>
<tr>
<td>YOFF 0.0E+0</td>
<td>This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOutpre:YOFF? query.)</td>
</tr>
<tr>
<td>YZERO 0.0E+0</td>
<td>This value indicates the vertical offset of the source waveform in units specified by the WFMOutpre:YUNIT command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YZERO? query.)</td>
</tr>
</tbody>
</table>
Example 6: Digital with 8 Bytes Per Point and MagniVu On

Goal: Transfer 25 points of Digital data from the oscilloscope to a PC using 8 bytes per point and Zoom on.

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DATa:SOUrce CH1_D1</td>
<td></td>
</tr>
<tr>
<td>:ACQuire:MAGnivu 1</td>
<td></td>
</tr>
<tr>
<td>:DATa:START 1</td>
<td></td>
</tr>
<tr>
<td>:DATa:STOP 25</td>
<td></td>
</tr>
<tr>
<td>:WFMOutpre:ENCdg ASCII</td>
<td></td>
</tr>
<tr>
<td>:WFMOutpre:BYT_Nr 8</td>
<td></td>
</tr>
<tr>
<td>:HEADer 1</td>
<td></td>
</tr>
<tr>
<td>:VERBose 1</td>
<td></td>
</tr>
<tr>
<td>:WFMOutpre?</td>
<td>Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the oscilloscope to a PC or other device (see next table for explanations):</td>
</tr>
<tr>
<td>:WFMOUTPRE:BYT_NR 8;BIT_NR 64;ENCDG ASCII;BN_FMT RI;BYT_OR MSB;WFID &quot;Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode&quot;;NR_PT 25;PT_FMT Y;PT_ORDER LINEAR;XUNIT &quot;s&quot;;XINCR 1.2121E-9;XZERO -6.0606E-6;PT_OFF 0;YUNIT &quot;State&quot;;YMULT 1.0000;YOFF 0.0E+0;YZERO 0.0E+0</td>
<td></td>
</tr>
<tr>
<td>:CURVe?</td>
<td>Returns the following values. Each value represents a data point:</td>
</tr>
<tr>
<td>:CURve 80000FB787,80000FB787,E8000FB787,8C000FB7E7,8C000FB7E7,84000FB7EF,CC000FB7A7,8C000FB7A7,8C000FB7A7,84000FB7A7,84000FB7A7,84000FB7A7,80000FB7A7,80000FB7A7,80000FB7A7,80000FB7A7,80000FB7A7,80000FB7A7,80000FB7A7,80000FB7A7,8C000FB7A7,8C000FB7A7,8C000FB7A7,8C000FB7A7,84000FB7EF,CC000FB7A7,8C000FB7A7,8C000FB7A7</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE.** The returned hexadecimal data values are formatted without leading zeroes.

**NOTE.** You can also use the WAVFrm? query, which concatenates the WFMOutpre? and CURVe? queries.
## Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples

<table>
<thead>
<tr>
<th>WFMOutpre? Query results</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFMOUTPRE:BYT_NR 8</td>
<td>This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the <code>WFMOutpre:BYT_NR</code> command. Note that changing this value automatically changes the <code>BIT_NR</code> value accordingly. (This value can also be set using the <code>DATa:WIDth</code> command.)</td>
</tr>
<tr>
<td>BIT_NR 64</td>
<td>This value specifies the number of bits per data point in the waveform data. To change this value, use the <code>WFMOutpre:BIT_Nr</code> command. Note that changing this value automatically changes the <code>BYT_NR</code> value accordingly.</td>
</tr>
<tr>
<td>ENCDG ASCII</td>
<td>This value specifies the encoding of the waveform data. To change this value (the other possibility is BINARY), use the <code>WFMOutpre:ENCdg</code> command. (This value can also be set using the <code>DATa:ENCdg</code> command, which provides the ability to set the <code>WFMOutpre:ENCdg</code>, <code>WFMOutpre:BN_Fmt</code>, and <code>WFMOutpre:BYT_Or</code> values using a single command.)</td>
</tr>
<tr>
<td>BN_FMT RI</td>
<td>This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the <code>WFMOutpre:BN_FMT</code> command. Note: this field is not applicable for ASCII encoding.</td>
</tr>
<tr>
<td>BYT_OR MSB</td>
<td>This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the <code>WFMOutpre:BYT_Or</code> command. Note: this field is not applicable for ASCII encoding.</td>
</tr>
<tr>
<td>WFID &quot;Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode&quot;</td>
<td>This quoted string represents information about the source waveform that would be returned by a <code>WFMOutpre:WFId?</code> query. It cannot be changed.</td>
</tr>
<tr>
<td>NR_PT 25</td>
<td>This value indicates the number of data points in the waveform record. (If you would like to determine only this value, use the <code>WFMOutpre:NR_Pt?</code> query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the <code>DATa:STARt</code> and <code>DATa:STOP</code> commands.</td>
</tr>
<tr>
<td>PT_FMT Y</td>
<td>This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only – the returned values can be Y for YT format or ENV for envelope format (min/max pairs). (If you would like to determine only this value, use the <code>WFMOutpre:PT_Fmt?</code> query.)</td>
</tr>
<tr>
<td>PT_ORDER LINEAR</td>
<td>This value is always LINEar.</td>
</tr>
<tr>
<td>XUNIT &quot;s&quot;</td>
<td>This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. Typically, this value is &quot;s&quot;, representing seconds. When using the math waveform as a source, the value can be &quot;s&quot; or &quot;Hz&quot;. This is query only. (If you would like to determine only this value, use the <code>WFMOutpre:XUNit?</code> query.)</td>
</tr>
<tr>
<td>XINCR 1.2121E-9</td>
<td>This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the <code>WFMOutpre:XINcr?</code> query.)</td>
</tr>
<tr>
<td>XZERO -6.0606E-6</td>
<td>This value indicates the time, in seconds, or frequency, in hertz, of the first data point in the waveform record. This time or frequency is relative to the time of the trigger, which is always 0. So, this XZERo time or frequency can be negative. This is query only. (If you would like to determine only this value, use the <code>WFMOutpre:XZERo?</code> query.)</td>
</tr>
<tr>
<td>PT_OFF 0</td>
<td>This is a query provided only for compatibility with other Tektronix oscilloscopes. The returned value is always 0. (If you would like to determine only this value, use the <code>WFMOutpre:PT_Off?</code> query.)</td>
</tr>
</tbody>
</table>
Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples

<table>
<thead>
<tr>
<th>WFMOutpre? Query results</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>YUNIT &quot;State&quot;</td>
<td>This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform—in this case, State. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNit? query.)</td>
</tr>
<tr>
<td>YMULT 1.0000</td>
<td>This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the WFMOutpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMUlt? query.)</td>
</tr>
<tr>
<td>YOFF 0.0E+0</td>
<td>This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOutpre:YOFf? query.)</td>
</tr>
<tr>
<td>YZERO 0.0E+0</td>
<td>This value indicates the vertical offset of the source waveform in units specified by the WFMOutpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YZero? query.)</td>
</tr>
</tbody>
</table>

Example 7: RF Frequency Domain Waveform

Goal: Transfer 5 points of RF frequency domain data from the oscilloscope to a PC using 4 bytes per point.

NOTE. A frequency domain waveform used in data transfer from the oscilloscope to a PC or other device may be one of the four RF frequency domain traces or the Spectrum Math waveform.

Command Comment
:DATa:SOURce RF_NORMal
:DATa:STARt 495
:DATa:STOP 505
:WFMOutpre:ENCdg ASCii
:WFMOutpre:BYT Nr 4
:HEAder 1
:VERBose 1
### Appendix D: Waveform Transfer (WFMOutpre and CURve Query) Examples

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>:WFMOutpre?</td>
<td>Returns the following values. Each value represents the current settings that a CURve? query will use to format the data that will be transferred from the oscilloscope to a PC or other device (see next table for explanations):</td>
</tr>
<tr>
<td></td>
<td>:WFMOUTPRE:BYT_NR 4;BIT_NR 32;ENCDG ASCII;BN_FMT FP;BYT_OR MSB;WFID &quot;RF_NORMAL, unknown coupling, 10.00W/div, 200.0MHz/div, 1001 points, Sample mode, Center Freq: 2.400GHz, Span: 2.000GHz, Reference Level: 1.000mW&quot;;NR_PT 21;PT_FMT Y;PT_ORDER LINEAR;XUNIT &quot;Hz&quot;;XINCR 2.0000E+6;XZERO 1.4000E+9;PT_OFF 0;YUNIT &quot;W&quot;;YMULT 1.5625E-3;YOFF 0.0E+0;YZERO 0.0E+0;DOMAIN FREQUENCY;WFMTYPE RF_FD;CENTERFREQUENCY 2.4000E+9;SPAN 2.0000E+9;REFLEVEL 1.0000E-3</td>
</tr>
<tr>
<td>:CURve?</td>
<td>Returns the following values. Each value represents a data point:</td>
</tr>
</tbody>
</table>

### WFMOutpre? Query results

| BYT_NR 4 | This value specifies the number of bytes per data point in the waveform data. This value is fixed for RF frequency domain traces. |
| BIT_NR 32 | This value specifies the number of bits per data point in the waveform data. This value is fixed for RF frequency domain traces. |
| ENCDG ASCII | This value specifies the encoding of the waveform data. To change this value (the other possibility for RF frequency domain traces is FPBINARY), use the WFMOutpre:ENCdg command. This value can also be set using the DATa:ENCdg command. |
| BN_FMT FP | This value specifies the binary format, which in this case is FP (floating point). This value is fixed for RF frequency domain traces. Note: This field is not applicable for ASCii encoding. |
| BYT_OR MSB | This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOutpre:BYT_Or command. Note: This field is not applicable for ASCii encoding. |
| WFID "RF_NORMAL, unknown coupling, 10.00W/div, 200.0MHz/div, 1001 points, Sample mode, Center Freq: 2.400GHz, Span: 2.000GHz, Reference Level: 1.000mW" | This quoted string represents information about the source waveform that would be returned by a WFMOutpre:WFId? Query. It cannot be changed. |
| NR_PT 21 | This value indicates the number of data points in the waveform record. (If you would like to determine only this value, use the WFMOutpre:NR_Pt? query.) Note: This value is typically equal to the full record length of the RF frequency domain trace, but you also have the option to transfer only a portion of the record using the DATa:STARt and DATa:STOP commands. |
| PT_FMT Y | This value indicates the format of the data points in the waveform record. In this case, the value represents YF (Amplitude vs. Frequency) format. This is query only. (If you would like to determine only this value, use the WFMOutpre:PT_FMT query. |
| PT_ORDER LINEAR | This value is always LINEar |
### Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XUNIT &quot;Hz&quot;</td>
<td>This value indicates the units of the x-axis of the waveform record. This value is always “Hz” for RF frequency domain traces. This is query only. (If you would like to determine only this value, use the WFMOutpre:XUNit? query.)</td>
</tr>
<tr>
<td>XINCR 2.0000E+6</td>
<td>This value indicates the frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOutpre:XINcr? query.)</td>
</tr>
<tr>
<td>XZERO 1.4000E+9</td>
<td>This value indicates the frequency, in hertz, of the first data point in the waveform record. This frequency is relative to the time of the trigger, which is always 0. This XZErO frequency can be negative. This is query only. (If you would like to determine only this value, use the WFMOutpre:XZEro? query.)</td>
</tr>
<tr>
<td>PT_OFF 0</td>
<td>This is a query provided only for compatibility with other Tektronix oscilloscopes. The returned value is always 0. (If you would like to determine only this value, use the WFMOutpre:PT_Off? query.)</td>
</tr>
<tr>
<td>YUNIT &quot;W&quot;</td>
<td>This value indicates the units of data points in the waveform record. This value depends on the vertical units of the source waveform – in this case, watts. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNit? query.)</td>
</tr>
<tr>
<td>YMULT 1.5625E-3</td>
<td>This value indicates the multiplying factor to convert the data point values to the units specified by the WFMOutpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMUlt? query.)</td>
</tr>
<tr>
<td>YOFF 0.0E+0</td>
<td>This value indicates the vertical position of the source waveform. This value is unused for RF frequency domain traces and is always 0.</td>
</tr>
<tr>
<td>YZERO 0.0E+0</td>
<td>This value indicates the vertical offset of the source waveform. This value is unused for RF frequency domain traces and is always 0.</td>
</tr>
<tr>
<td>DOMAIN FREQUENCY</td>
<td>This value indicates the domain in which the source waveform is displayed and stored. For RF time domain traces, the domain is Time and waveform transfer information is treated as integer information. For RF frequency domain traces, the domain is Frequency and waveform transfer information is treated as floating point information. This is query only. (If you would like to determine only this value, use the WFMOutpre:DOMain? query.)</td>
</tr>
<tr>
<td>WFMTYPE RF_FD</td>
<td>This value indicates the type of the source waveform. RF_FD indicates an RF frequency domain trace (frequency domain waveform). This is query only. (If you would like to determine only this value, use the WFMOutpre:WFMTYPE? query.)</td>
</tr>
<tr>
<td>CENTERFREQUENCY 2.4000E+9</td>
<td>This value indicates the center frequency, in hertz, of the source waveform. This is query only. (If you would like to determine only this value, use the WFMOutpre:CENTERFREQuency? query.)</td>
</tr>
<tr>
<td>SPAN 2.0000E+9</td>
<td>This value indicates the frequency span, in hertz, of the source waveform. This is query only. (If you would like to determine only this value, use the WFMOutpre:SPAN? query.)</td>
</tr>
<tr>
<td>REFLEVEL 1.0000E-3</td>
<td>This value indicates the reference level, in watts, of the source waveform. This is query only. (If you would like to determine only this value, use the WFMOutpre:REFLEvel? query.)</td>
</tr>
</tbody>
</table>
Appendix E: Search and Trigger Command Sequence Examples

The following are some example command sequences that show a variety of different searches and triggers. The commands in these sequences are not order-dependent.

To use these examples, connect channel 1, channel 2, channel 3 and channel 4 to the probe compensation signal located on the right hand side of the front panel.

The search and trigger command group sections contain more information on general search and trigger concepts.

Example 1: Single Threshold Edge Search

Goal: Search the channel 2 waveform and place a mark at each instance where it crosses below a threshold of 1.4 volts.

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>*RST;:*OPC?</td>
<td>Resets the oscilloscope and waits for that operation to complete (approximately 1 to 5 seconds depending on the complexity of the previous setup).</td>
</tr>
<tr>
<td>display:waveview1:ch2:state 1</td>
<td>Turns the CH2 waveform on.</td>
</tr>
<tr>
<td>:AUTOset EXECute&gt;;:*OPC?</td>
<td>Autosets the displayed waveform CH2 and waits for the auto setup to complete.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGger:A:TYPe EDGE</td>
<td>Specifies that this will be an edge search (a mark will be placed when the source waveform passes through a specified threshold level in the specified direction).</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGger:A:EDGE:SOURce CH2</td>
<td>Specifies the CH2 waveform as the source waveform.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGger:A:LOWerthreshold:CH2 1.4</td>
<td>Specifies 1.4 volts as the threshold level.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGger:A:EDGE:SLOpe FALL</td>
<td>Specifies the falling edge as the direction.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:STATE 1</td>
<td>Turns the search on.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TOTal?</td>
<td>Returns 4, indicating that CH2 fell below the 1.4 volt threshold 4 times.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:LIST?</td>
<td>Returns a list of the 4 marks:</td>
</tr>
</tbody>
</table>

CH2, 11.2411, 11.2411, 11.2411, -1.5504E-3, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, CH2, 37.0737, 37.0737, 37.0737, -517.0517E-6, 0.0E+0, 0.0E+0, 0.0E+0, CH2, 62.9163, 62.9163, 62.9163, 516.6517E-6, 0.0E+0, 0.0E+0, 0.0E+0, CH2, 88.7489, 88.7489, 88.7489, 1.5500E-3, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0 |
Appendix E: Search and Trigger Command Sequence Examples

Example 2: Single Threshold Edge Trigger

Goal: Trigger on the channel 2 waveform when the waveform crosses below a threshold of 1.4 volts.

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>*RST</td>
<td>Resets the oscilloscope. Wait for the reset to complete (approximately 3 seconds).</td>
</tr>
<tr>
<td>:DISPLAY:WAVEVIEW1:CH2:STATE 1</td>
<td>Turns the CH2 waveform on.</td>
</tr>
<tr>
<td>:TRIgger:A:TYPe EDGE</td>
<td>Specifies that this will be an edge trigger (trigger will occur when the source waveform passes through a specified threshold level in the specified direction).</td>
</tr>
<tr>
<td>:TRIgger:A:EDGE:SOUrce CH2</td>
<td>Specifies the CH2 waveform as the source waveform.</td>
</tr>
<tr>
<td>:TRIgger:A:LOWerthreshold:CH2 1.4</td>
<td>Specifies 1.4 volts as the threshold level.</td>
</tr>
<tr>
<td>:TRIgger:A:EDGE:SLOpe FALL</td>
<td>Specifies as falling edge as the direction.</td>
</tr>
<tr>
<td>:TRIgger:STATE?</td>
<td>Should return TRIGGER, not AUTO.</td>
</tr>
</tbody>
</table>

Example 3: Dual Threshold Runt Search

Goal: Search the channel 3 waveform for negative runt pulses and place a mark at each instance when the waveform drops below an upper threshold of 1.4 volts, but does not cross a lower threshold of -2 volts before re-crossing the upper threshold. The pulse width must be less than 600 microseconds.

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>*RST</td>
<td>Resets the oscilloscope. Wait for the reset to complete (approximately 3 seconds).</td>
</tr>
<tr>
<td>:DISPLAY:WAVEVIEW1:CH3:STATE 1</td>
<td>Turns the CH3 waveform on.</td>
</tr>
<tr>
<td>:AUTOset EXECute</td>
<td>Autosets the displayed waveform CH3. Wait for the autoset to complete (approximately 3 seconds).</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIgger:A:TYPe RUNT</td>
<td>Specifies that this will be a runt search (a mark will be placed on a pulse amplitude that crosses one threshold but fails to cross a second threshold before re-crossing the first).</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIgger:A:EDGE:SOUrce CH3</td>
<td>Specifies to use channel 3 as the source waveform.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIgger:A:LOWerthreshold:CH3 -2</td>
<td>Specifies to use -2 volts as the lower threshold.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIgger:A:UPPerthreshold: CH3 1.4</td>
<td>Specifies to use 1.4 volts as the upper threshold.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIgger:A:RUNT:POLarity NEGative</td>
<td>Specifies to search for when the runt polarity is negative.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIgger:A:RUNT:WHEn LESSthan</td>
<td>Specifies to search for when the pulse width is less than 600E-6 seconds.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:STATE 1</td>
<td>Turns the search on.</td>
</tr>
</tbody>
</table>
### Appendix E: Search and Trigger Command Sequence Examples

#### Command Comment

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH1:TOTal?</td>
<td>Returns 3, indicating that 3 total negative runt pulses were less than 600E-6 seconds wide</td>
</tr>
</tbody>
</table>
| SEARCH:SEARCH1:LIST? | Returns a list of 3 marks: 

CH3,11.2311,24.1624,24.1624,1.0335E3,0.0E+0,0.0E+0,0.0E+0;CH3,37.0837,50.0150,60.0600E9,0.0E+0,0.0E+0,0.0E+0;CH3,62.9263,75.8576,75.8576,1.0343E-3,0.0E+0,0.0E+0,0.0E+0 |

**NOTE.** You could use a similar command sequence with a transition type search.

### Example 4: Single Threshold Logic Search on Three Waveforms

**Goal:** Search the channel 1, 2 and 3 waveforms and place a mark at each instance when either channel 1 is above 1.4 volts, channel 2 is above 1.5 volts, or channel 3 is above 1.3 volts.

#### Command Comment

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>*RST</td>
<td>Resets the oscilloscope. Wait for the reset to complete (approximately 3 seconds).</td>
</tr>
<tr>
<td>:DISPLAY:WAVEVIEW1:CH1:STATE 1</td>
<td>Turns the CH1 waveform on.</td>
</tr>
<tr>
<td>:DISPLAY:WAVEVIEW1:CH2:STATE 1</td>
<td>Turns the CH2 waveform on.</td>
</tr>
<tr>
<td>:DISPLAY:WAVEVIEW1:CH3:STATE 1</td>
<td>Turns the CH3 waveform on.</td>
</tr>
<tr>
<td>:AUTOset EXECute</td>
<td>Autosets the displayed waveforms CH1, CH2 and CH3. Wait for the autoset to complete (approximately 3 seconds).</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGger:A:TYPe LOGIc</td>
<td>Specifies that this will be a logic search (a mark will be placed when all channels transition to the specified state).</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGger:A:LOGIc:INPut:CH1 HIGH</td>
<td>Specifies the Boolean logic criteria for channel 1; in this case, high.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGger:A:LOGIc:INPut:CH2 HIGH</td>
<td>Specifies the Boolean logic criteria for channel 2; in this case, high.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGger:A:LOGIc:INPut:CH3 HIGH</td>
<td>Specifies the Boolean logic criteria for channel 3; in this case, high.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGger:A:LOGIc:THReshold:CH1 1.4</td>
<td>Specifies to use 1.4 volts as the threshold for CH1.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGger:A:LOGIc:THReshold:CH2 1.5</td>
<td>Specifies to use 1.5 volts as the threshold for CH2.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGger:A:LOGIc:THReshold:CH3 1.3</td>
<td>Specifies to use 1.3 volts as the threshold for CH3.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGger:A:LOGIc:PATtern:WHen TRUE</td>
<td>Specifies the condition for generating a logic pattern search; in this case, true.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIGger:A:LOGIc:FUNCtion OR</td>
<td>Specifies the logic operator for the logic search; in this case, OR.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:STATE 1</td>
<td>Turns the search on.</td>
</tr>
</tbody>
</table>
### Appendix E: Search and Trigger Command Sequence Examples

<table>
<thead>
<tr>
<th>Command</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH1:TOTal?</td>
<td>Returns 3 or 4, indicating 3 or 4 times when one of channels 1, 2 or 3 became high. Note: Depending upon the number of transitions displayed, you may get 3 or 4 search marks for this example. If you get 3 search marks, try adjusting the horizontal position until you see 4 search marks.</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:LIST?</td>
<td>Returns a list of 3 marks on 3 waveforms:</td>
</tr>
<tr>
<td></td>
<td>CH1, 24.1600, 24.1600, 24.1600, 24.1600, 1.0336E+3, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0</td>
</tr>
<tr>
<td></td>
<td>CH2, 24.1600, 24.1600, 24.1600, 24.1600, 1.0336E+3, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0</td>
</tr>
<tr>
<td></td>
<td>CH3, 24.1600, 24.1600, 24.1600, 24.1600, 1.0336E+3, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0</td>
</tr>
<tr>
<td></td>
<td>CH1, 50.0000, 50.0000, 50.0000, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0</td>
</tr>
<tr>
<td></td>
<td>CH2, 50.0000, 50.0000, 50.0000, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0</td>
</tr>
<tr>
<td></td>
<td>CH3, 50.0000, 50.0000, 50.0000, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0</td>
</tr>
<tr>
<td></td>
<td>CH1, 75.8300, 75.8300, 75.8300, 1.0332E-3, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0</td>
</tr>
<tr>
<td></td>
<td>CH2, 75.8300, 75.8300, 75.8300, 1.0332E+3, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0</td>
</tr>
<tr>
<td></td>
<td>CH3, 75.8300, 75.8300, 75.8300, 1.0332E+3, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0</td>
</tr>
<tr>
<td></td>
<td>CH1, 75.8300, 75.8300, 75.8300, 1.0332E-4, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0</td>
</tr>
<tr>
<td></td>
<td>CH2, 75.8300, 75.8300, 75.8300, 1.0332E-4, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0</td>
</tr>
<tr>
<td></td>
<td>CH3, 75.8300, 75.8300, 75.8300, 1.0332E-4, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0</td>
</tr>
</tbody>
</table>

E-4
ASCII
Acronym for the American Standard Code for Information Interchange. Controllers transmit commands to the instrument using ASCII character encoding.

Address
A 7-bit code that identifies an instrument on the communication bus. The digitizing instrument must have a unique address for the controller to recognize and transmit commands to it.

Backus-Naur Form (BNF)
A standard notation system for command syntax diagrams. The syntax diagrams in this manual use BNF notation.

Controller
A computer or other device that sends commands to and accepts responses from the digitizing instrument.

EOI
A mnemonic referring to the control line End or Identify. One of the two possible end-of-message terminators.

EOM
A generic acronym referring to the end-of-message terminator. The end-of-message terminator can be either an EOI or the ASCII code for line feed (LF).

Equivalent-Time sampling (ET)
A sampling mode in which the instrument acquires signals over many repetitions of the event. This instrument uses a type of equivalent time sampling called random equivalent time sampling. It utilizes an internal clock that runs asynchronously with respect to the input signal and the signal trigger. The instrument takes samples continuously, independent of the trigger position, and displays them based on the time difference between the sample and the trigger. Although the samples are taken sequentially in time, they are random with respect to the trigger.

Real-Time sampling
A sampling mode where the instrument samples fast enough to completely fill a waveform record from a single trigger event. Use real-time sampling to capture single-shot or transient events.

IEEE
An acronym for the Institute for Electrical and Electronic Engineers.

Serial poll
A device on the bus can request service from the Controller by asserting the SRQ line. When a controller acknowledges the SRQ, it serial polls each device on the bus to determine which device on the bus requested service. Any device requesting service returns a status byte indicating it needs to be serviced and then
unasserts the SRQ line. Devices not requiring service return a status byte that indicates they do not need servicing.

**TEKSecure**
A Tektronix custom command that initializes both waveform and setup memories. This overwrites any previously stored data.
Index

A
ACQuire?, 2-107
ACQuire:FASTAcq:PALEtte, 2-107
ACQuire:FASTAcq:STATE, 2-108
ACQuire:MAXSamplerate?, 2-108
ACQuire:MODE, 2-108
ACQuire:NUMAcq?, 2-110
ACQuire:NUMAv, 2-110
ACQuire:NUMFRAMESACQuired?, 2-111
ACQuire:SEQUence:CURrent?, 2-111
ACQuire:SEQUence:MODe, 2-111
ACQuire:SEQUence:NUMSEQUence, 2-112
ACQuire:STATE, 2-112
ACQuire:STOPAfter, 2-113
AFG:AMPLitude, 2-113
AFG:ARBitrary:SOUrce, 2-114
AFG:FREQuency, 2-114
AFG:FUNCtion, 2-115
AFG:HIGHLevel, 2-116
AFG:LOWLevel, 2-116
AFG:NOISEAdd:PERCent, 2-117
AFG:NOISEAdd:STATE, 2-117
AFG:OFFSet, 2-118
AFG:OUTPut:LOAd:IMPEDance, 2-118
AFG:OUTPut:STATE, 2-119
AFG:PERIod, 2-119
AFG:PULse:WIDth, 2-120
AFG:RAMP:SYMmetry, 2-120
AFG:SQUare:DUty, 2-121
ALIas, 2-121
ALIas:CATalog?, 2-122
ALIas:DEFine, 2-122
ALIas:DELETe, 2-123
ALIas:DELETe:ALL, 2-123
ALIas:DELETe:NAME, 2-124
ALIas:STATE, 2-124
ALLEv?, 2-125
AUTOSAVE:EPITIMEOUT, 2-125
AUTOSAVE:EUITIMEOUT, 2-126
AUTOSet, 2-126
AUTOSet:ACQuisition:ENAble, 2-127
AUTOSet:ENAble, 2-127
AUTOSet:HORizontal:ENAble, 2-128
AUTOSet:TRIGger:ENAble, 2-128
AUTOSet:DATEM:ENAble, 2-129
AUTOSet:VERTical:ENABLe, 2-130
AUTOSet:VERTical:OPTIMize, 2-130
AUXout:EDGE, 2-130
AUXout:SOUrce, 2-131

B
BUS:ADDNew, 2-131
BUS:B<x>:ARINC429A:BITRate, 2-132
BUS:B<x>:ARINC429A:BITRate:CUSTom, 2-132
BUS:B<x>:ARINC429A:DATAFormat, 2-133
BUS:B<x>:ARINC429A:POLARity, 2-134
BUS:B<x>:ARINC429A:SOUrce, 2-134
BUS:B<x>:ARINC429A:THRESHold, 2-135
BUS:B<x>:AUDio:BITDelay, 2-136
BUS:B<x>:AUDio:BITOrder, 2-136
BUS:B<x>:AUDio:POLARity, 2-137
BUS:B<x>:AUDio:SOURce, 2-137
BUS:B<x>:AUDio:THRESHold, 2-138
BUS:B<x>:AUDio:DATA:POLARity, 2-139
BUS:B<x>:AUDio:DATA:SIZE, 2-139
BUS:B<x>:AUDio:DATA:SOURce, 2-140
BUS:B<x>:AUDio:DATA:THRESHold, 2-141
BUS:B<x>:AUDio:DATA:WORDSizE, 2-141
BUS:B<x>:AUDio:FRAME:CLOCKBITSPERCHANNEL, 2-142
BUS:B<x>:AUDio:FRAME:SIZE, 2-142
BUS:B<x>:AUDio:TYPE, 2-143
BUS:B<x>:AUDio:WORDSizE:POLARity, 2-144
BUS:B<x>:AUDio:WORDSizE:SOURce, 2-144
BUS:B<x>:AUDio:WORDSizE:THRESHold, 2-145
BUS:B<x>:CAN:BITRate, 2-146
BUS:B<x>:CAN:BITRate:VALue, 2-146
BUS:B<x>:CAN:FD:BITRate, 2-147
BUS:B<x>:CAN:FD:BITRate:CUSToM, 2-148
BUS:B<x>:CAN:SAMPLEpoint, 2-148
BUS:B<x>:CAN:SIGNAl:DATA, 2-149
BUS:B<x>:CAN:SOURce, 2-149
BUS:B<x>:CAN:STANDARD, 2-150
BUS:B<x>:CAN:THRESHold, 2-150
BUS:B<x>:DISplay:FORMAT, 2-151
BUS:B<x>:DIsplay:LAYout, 2-152
BUS:B<x>:ETHERnet:DATAMINUS:THRESHold, 2-152
Index-2 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
BUS:<x>: SPMI: SCLk: THReshold, 2-205
BUS:<x>: SPMI: SDA: SOURce, 2-206
BUS:<x>: SPMI: SDA: THReshold, 2-207
BUS:<x>: TYPE, 2-207
BUS:<x>: USB: BITRate, 2-208
BUS:<x>: USB: DATAMINUSTHRESHold, 2-209
BUS:<x>: USB: DATAPLUSTHRESHold, 2-210
BUS:<x>: USB: LOWTHRESHold, 2-210
BUS:<x>: USB: SIGNALTYpe, 2-211
BUS:<x>: USB: SOURce, 2-211
BUS:<x>: USB: SOURce:DMINus, 2-212
BUS:<x>: USB: SOURce: DPLUs, 2-213
BUS:<x>: USB: THRESHold, 2-214
BUS:DELETE, 2-214
BUS:LIST?, 2-215
BUSTABLE: ADDNew, 2-215
BUSTABLE: DELete, 2-215
BUSTABLE: LIST?, 2-216
BUSY?, 2-216

C

*CAL?, 2-217
CALibrate?, 2-217
CALibrate INTErnal, 2-218
CALibrate INTErnal: START, 2-218
CALibrate INTErnal: STATus?, 2-219
CALibrate PWRIUpstatus?, 2-219
CH<n>: PRObe: STATus?, 2-220
CH<x>?, 2-221
CH<x>: BANdwidth, 2-221
CH<x>: CLIPping?, 2-222
CH<x>: COUPling, 2-222
CH<x>: LABel: COLor, 2-246
CH<x>: LABel: FONT: BOLD, 2-246
CH<x>: LABel: FONT: ITALic, 2-247
CH<x>: LABel: FONT: SIZE, 2-247
CH<x>: LABel: FONT: TYPE, 2-248
CH<x>: LABel: FONT: UNDERline, 2-248
CH<x>: LABel: NAME, 2-249
CH<x>: DALL: LABel: COLor, 2-242
CH<x>: DALL: LABel: FONT: BOLD, 2-243
CH<x>: DALL: LABel: FONT: ITALic, 2-243
CH<x>: DALL: LABel: FONT: SIZE, 2-244
CH<x>: DALL: LABel: FONT: TYPE, 2-244
CH<x>: DALL: LABel: FONT: UNDERline, 2-245
CH<x>: DALL: LABel: NAME, 2-245
CH<x>: DESKeW, 2-223
CH<x>: LABel: COLor, 2-223
CH<x>: LABel: FONT: BOLD, 2-224
CH<x>: LABel: FONT: ITALic, 2-224
CH<x>: LABel: FONT: SIZE, 2-225
CH<x>: LABel: FONT: TYPE, 2-225
CH<x>: LABel: FONT: UNDERline, 2-226
CH<x>: LABel: NAME, 2-226
CH<x>: LABel: XPOS, 2-227
CH<x>: LABel: YPOS, 2-227
CH<x>: OFFSet, 2-228
CH<x>: POSition, 2-228
CH<x>: PRObe?, 2-229
CH<x>: PRObe AUTOZero, 2-229
CH<x>: PRObe DEGAUSS, 2-230
CH<x>: PRObe DEGAUSS: STATE?, 2-230
CH<x>: PRObe FORCEDRange, 2-231
CH<x>: PRObe GAIN?, 2-231
CH<x>: PRObe ID?, 2-231
CH<x>: PRObe ID: SERIALnumber?, 2-232
CH<x>: PRObe ID: TYPE?, 2-232
CH<x>: PRObe INPUTMode, 2-233
CH<x>: PRObe INPUTMode: AOFFSet, 2-233
CH<x>: PRObe INPUTMode: BOFFSet, 2-234
CH<x>: PRObe INPUTMode: COFFSet, 2-234
CH<x>: PRObe INPUTMode: DOFFSet, 2-235
CH<x>: PRObe RESistance?, 2-236
CH<x>: PRObe SET, 2-236
CH<x>: PRObe UNIts?, 2-236
CH<x>: PROBE Cal?, 2-237
CH<x>: PROBE Control, 2-237
CH<x>: PROBE Func: EXTAtten, 2-238
CH<x>: PROBE Func: EXTDAtten, 2-239
CH<x>: PROBE Func: EXTUnits, 2-239
CH<x>: PROBE Func: EXTUnits: STATE, 2-240
CH<x>: SCALE, 2-240
CH<x>: SCALeRA: Tio, 2-241
CH<x>: TERMINation, 2-241
CH<x>: VTERm BIAS, 2-242
CLEAR, 2-249
*CLS, 2-250
Command Groups, 2-11
CONFIGuration ANALOG: BANDWidth?, 2-250
CURVe, 2-251

D

DATa, 2-253
DATa ENCdg, 2-254
DISplay PLOTView<x>: CURSor FUNCTION, 2-290
DISplay PLOTView<x>: CURSor HBArs:
  APOSition, 2-291
DISplay PLOTView<x>: CURSor HBArs:
  AUNits?, 2-291
DISplay PLOTView<x>: CURSor HBArs:
  BPOSition, 2-292
DISplay PLOTView<x>: CURSor HBArs:
  BUNits?, 2-292
DISplay PLOTView<x>: CURSor HBArs:
  DELTa?, 2-293
DISplay PLOTView<x>: CURSor HBArs:
  MODE, 2-293
DISplay PLOTView<x>: CURSor HBArs:
  ONEOVERDELTATVALUE?, 2-294
DISplay PLOTView<x>: CURSor SCREEN:
  AXPOSition, 2-294
DISplay PLOTView<x>: CURSor SCREEN:
  AYPOSition, 2-295
DISplay PLOTView<x>: CURSor SCREEN:
  BXPOSition, 2-295
DISplay PLOTView<x>: CURSor SCREEN:
  BYPOSition, 2-296
DISplay PLOTView<x>: CURSor:
  SPLITMODE, 2-296
DISplay PLOTView<x>: CURSor STATE, 2-297
DISplay PLOTView<x>: CURSor VBArS:
  APOSition, 2-297
DISplay PLOTView<x>: CURSor VBArS:
  BPOSition, 2-298
DISplay PLOTView<x>: CURSor VBArS:
  DELTa?, 2-298
DISplay PLOTView<x>: CURSor VBArS:
  UNits?, 2-299
DISplay PLOTView<x>: CURSor WAVEform:
  APOSition, 2-299
DISplay PLOTView<x>: CURSor WAVEform:
  BPOSition, 2-300
DISplay PLOTView<x>: GRIDlines, 2-300
DISPLAY PLOTView<x>: XAXIS SCALE, 2-301
DISPLAY PLOTView<x>: YAXIS SCALE, 2-301
DISplay PLOTView<x>: ZOOM XAXIS:
  FROM, 2-302
DISplay PLOTView<x>: ZOOM XAXIS: TO, 2-302
DISplay PLOTView<x>: ZOOM YAXIS:
  FROM, 2-303
DISplay PLOTView<x>: ZOOM YAXIS: TO, 2-303
DISplay REFFFTView<x>: CURSor:
  ASOUrce?, 2-305
DISplay REFFFTView<x>: CURSor:
  BSOUrce?, 2-305
DISplay REFFFTView<x>: CURSor DDT?, 2-305
DISplay REFFFTView<x>: CURSor:
  FUNCtion, 2-306
DISplay REFFFTView<x>: CURSor HBArS:
  APOSition, 2-306
DISplay REFFFTView<x>: CURSor HBArS:
  BPOSition, 2-307
DISplay REFFFTView<x>: CURSor HBArS:
  DELTa?, 2-308
DISplay REFFFTView<x>: CURSor HBArS:
  ONEOVERDELTATVALUE?, 2-309
DISplay REFFFTView<x>: CURSor:
  SPLITMODE, 2-311
DISplay REFFFTView<x>: CURSor SCREEN:
  AXPOSition, 2-311
DISplay REFFFTView<x>: CURSor SCREEN:
  AYPOSition, 2-312
DISplay REFFFTView<x>: CURSor SCREEN:
  BXPOSition, 2-312
DISplay REFFFTView<x>: CURSor SCREEN:
  BYPOSition, 2-313
DISplay REFFFTView<x>: CURSor:
  SPLITMODE, 2-313
DISplay REFFFTView<x>: CURSor VBArS:
  APOSition, 2-313
DISplay REFFFTView<x>: CURSor VBArS:
  BPOSition, 2-314
DISplay REFFFTView<x>: CURSor VBArS:
  DELTa?, 2-314
DISplay REFFFTView<x>: CURSor VBArS:
  UNits?, 2-315
DISplay REFFFTView<x>: CURSor WAVEform:
  APOSition, 2-315
DISplay REFFFTView<x>: CURSor WAVEform:
  BPOSition, 2-315
Index

DISPLAY REFFFTView<x>: CURSOR WAVEm-form:
BPOSition, 2-316
DISPLAY REFFFTView<x>: CURSOR WAVEm-form:
BVPOSition?, 2-316
DISPLAY REFFFTView<x>: GRIDlines, 2-317
DISPLAY REFFFTView<x>: REF:REF<x>:
STATE, 2-317
DISPLAY REFFFTView<x>: XAXIS: SCALE, 2-318
DISPLAY REFFFTView<x>: ZOOM: XAXIS:
FROM, 2-318
DISPLAY REFFFTView<x>: ZOOM: XAXIS: TO, 2-319
DISPLAY REFFFTView<x>: ZOOM: YAXIS: FROM, 2-319
DISPLAY REFFFTView<x>: ZOOM: YAXIS: TO, 2-320
DISPLAY REFFFTView<x>: BUS:B<x>:STATE, 2-323
DISPLAY REFFFTView<x>: BUS:B<x>:VERTical:
POSition, 2-324
DISPLAY REFFFTView<x>: CH<x>_D<x>:
STATE, 2-327
DISPLAY REFFFTView<x>: CH<x>_DALL:
STATE, 2-326
DISPLAY REFFFTView<x>: CH<x>_DALL:VERTical:
POSition, 2-327
DISPLAY REFFFTView<x>: CH<x>:STATE, 2-324
DISPLAY REFFFTView<x>: CH<x>:VERTical:
POSition, 2-325
DISPLAY REFFFTView<x>: CH<x>:VERTical:
SCALE, 2-325
DISPLAY REFFFTView<x>: CURSOR?, 2-328
DISPLAY REFFFTView<x>: CURSOR: CURSOR<x>?, 2-328
DISPLAY REFFFTView<x>: CURSOR: CURSOR<x>:
ASOURce, 2-339
DISPLAY REFFFTView<x>: CURSOR: CURSOR<x>:
BSOURce, 2-330
DISPLAY REFFFTView<x>: CURSOR: CURSOR<x>:
DDT?, 2-330
DISPLAY REFFFTView<x>: CURSOR: CURSOR<x>:
FUNCTION, 2-331
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
HBArS APOSition, 2-331
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
HBArS AUNIts?, 2-332
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
HBArS BPOSition, 2-333
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
HBArS BUNIts?, 2-333
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
HBArS DELTa?, 2-334
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
MODE, 2-334
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
ONEOVERDELTATVALUE?, 2-335
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
SCREEN AXPOsition, 2-335
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
SCREEN AYPOsition, 2-336
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
SCREEN BXPOsition, 2-337
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
SCREEN BYPOsition, 2-337
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
SPLITMODE, 2-338
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
STATE, 2-338
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
VBArS APOSition, 2-339
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
VBArS BPOSition, 2-340
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
VBArS DELTa?, 2-340
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
VBArS UNIts?, 2-341
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
WAVEform APOsition, 2-341
DISPLAY WAVEmView<x>: CURSOR: CURSOR<x>:
WAVEform BPOsition, 2-342
DISPLAY WAVEmView<x>: FILTER, 2-342
DISPLAY WAVEmView<x>: GRATicule, 2-343
DISPLAY WAVEmView<x>: INTENSITy:
GRATicule, 2-343
DISPLAY WAVEmView<x>: INTENSITy:
WAVEform, 2-344
DISPLAY WAVEmView<x>: MATH:MATH<x>:
AUTOScale, 2-344

Index-6 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
FILESystem: READFile, 2-382
FILESystem: RENAME, 2-382
FILESystem: RMDIR, 2-383
FILESystem: UNMOUNT: DRIVE, 2-384
FILESystem: WRITEFile, 2-384
FPANEL: PRESS, 2-384
FPANEL: TURN, 2-385
HEADER, 2-385
HORIZONTAL?, 2-386
HORIZONTAL: ACQDURATION?, 2-387
HORIZONTAL: DELAY: MODE, 2-387
HORIZONTAL: DELAY: TIME, 2-388
HORIZONTAL: DIVISIONS?, 2-388
HORIZONTAL: FASTframe?, 2-389
HORIZONTAL: FASTframe COUNT, 2-389
HORIZONTAL: FASTframe MAXFRAMES?, 2-389
HORIZONTAL: FASTframe MULTIPLEFRAMES:
   MODE, 2-390
HORIZONTAL: FASTframe REF: FRAME, 2-390
HORIZONTAL: FASTframe REF: INCLUDE, 2-391
HORIZONTAL: FASTframe SELECTED?, 2-391
HORIZONTAL: FASTframe STATE, 2-391
HORIZONTAL: FASTframe SUMFrame?, 2-392
HORIZONTAL: FASTframe SUMFrame: STATE, 2-392
HORIZONTAL: FASTframe TIME: STAMP: ALL?, 2-393
HORIZONTAL: FASTframe TIME: STAMP: DELTA?, 2-393
HORIZONTAL: FASTframe TIME: STAMP: REFERENCE?, 2-394
HORIZONTAL: FASTframe TIME: STAMP: SELECTED?, 2-394
HORIZONTAL: FASTframe: XZERo ALL?, 2-394
HORIZONTAL: FASTframe: XZERo REF, 2-395
HORIZONTAL: FASTframe: XZERo SELECTED?, 2-395
HORIZONTAL: MAXIMUM: VALUE, 2-397
HORIZONTAL: MODE: AUTOMATIC: FASTAQ:
   RECORDLENGTH: MAXIMUM: VALUE, 2-397
HORIZONTAL: MODE: AUTOMATIC:
   FASTAQ: RECORDLENGTH: MAXIMUM: ZOOMOVERRIDE, 2-397
HORIZONTAL: MODE: MANUAL: CONFIGURE, 2-398
HORIZONTAL: MODE: RECORDLENGTH, 2-398
HORIZONTAL: MODE: SAMPLERATE, 2-399
HORIZONTAL: SCALe, 2-400
HORIZONTAL: PREVIEWSTATE?, 2-400
HORIZONTAL: RECORDLENGTH, 2-400
HORIZONTAL: ROLL?, 2-401
HORIZONTAL: SAMPLERATE, 2-401
HORIZONTAL: SAMPLERATE: ANALYZEMODE:
   MINIMUM: OVERRIDE, 2-402
HORIZONTAL: SAMPLERATE: ANALYZEMODE:
   MINIMUM: VALUE, 2-402
HORIZONTAL: SCALe, 2-403
ID?, 2-403
*IDN?, 2-404
LIC: UNINSTALL?, 2-404
LICENSE?, 2-405
LICENSE: APPID?, 2-405
LICENSE: COUNT?, 2-405
LICENSE ERROR?, 2-406
LICENSE: GMT?, 2-406
LICENSE: HID?, 2-407
LICENSE: INSTALL, 2-407
LICENSE: ITEM?, 2-407
LICENSE: LIST?, 2-408
LICENSE: VALIDATE?, 2-408
LOCK, 2-409
*LRN?, 2-410
MAINWINDOW: RRB DISPLAYSTATE, 2-411
MATH: ADDNEW, 2-411
MATH: DELETE, 2-412
MATH: LIST?, 2-412
MATH: MATH<xs>: AVG: MODE, 2-412
MATH: MATH<xs>: AVG: WEIGHT, 2-413
MATH: MATH<xs>: DEFINE, 2-414
MATH: MATH<xs>: FUNCTION, 2-414
MATH: MATH<xs>: GATING, 2-415
MATH: MATH<xs>: LABEL: COLOR, 2-415
MATH: MATH<xs>: LABEL: FONT: BOLD, 2-416
MATH: MATH<xs>: LABEL: FONT: ITALIC, 2-416
MATH: MATH<xs>: LABEL: FONT: SIZE, 2-417
MATH: MATH<xs>: LABEL: FONT: TYPE, 2-417
MATH: MATH<xs>: LABEL: FONT: UNDERLINE, 2-418
MATH: MATH<xs>: LABEL: NAME, 2-418
MATH: MATH<x>: LABEL: XPOS, 2-419
MATH: MATH<x>: LABEL: YPOS, 2-419
MATH: MATH<x>: SOURCE<x>, 2-420
MATH: MATH<x>: SPECTRAL: HORIZ, 2-420
MATH: MATH<x>: SPECTRAL: MAG, 2-421
MATH: MATH<x>: SPECTRAL: PHASE, 2-421
MATH: MATH<x>: SPECTRAL: SOURCE, 2-422
MATH: MATH<x>: SPECTRAL: SUPPRESS, 2-423
MATH: MATH<x>: SPECTRAL: VALUE, 2-423
MATH: MATH<x>: SPECTRAL: TYPE, 2-424
MATH: MATH<x>: SPECTRAL: UNWRAP, 2-424
MATH: MATH<x>: SPECTRAL: UNWRAP: DEGREES, 2-425
MATH: MATH<x>: SPECTRAL: WINDOW, 2-425
MATH: MATH<x>: TYPE, 2-426
MATH: MATH<x>: VUNIT, 2-427
MEASUREMENT?, 2-428
MEASUREMENT: ADDMEAS, 2-429
MEASUREMENT: ADDNEW, 2-436
MEASUREMENT: ANNOTATE, 2-436
MEASUREMENT: AUTOSET, 2-437
MEASUREMENT: CH<x>: REFLEVELS: ABsolute: FALLHIGH, 2-437
MEASUREMENT: CH<x>: REFLEVELS: ABsolute: FALLLOW, 2-438
MEASUREMENT: CH<x>: REFLEVELS: ABsolute: FALLMID, 2-438
MEASUREMENT: CH<x>: REFLEVELS: ABsolute: HYSTERESIS, 2-439
MEASUREMENT: CH<x>: REFLEVELS: ABsolute: RISEHIGH, 2-439
MEASUREMENT: CH<x>: REFLEVELS: ABsolute: RISELow, 2-440
MEASUREMENT: CH<x>: REFLEVELS: ABsolute: RISEMID, 2-440
MEASUREMENT: CH<x>: REFLEVELS: BASETop, 2-441
MEASUREMENT: CH<x>: REFLEVELS: METHOD, 2-442
MEASUREMENT: CH<x>: REFLEVELS: PERCENT: FALLHIGH, 2-443
MEASUREMENT: CH<x>: REFLEVELS: PERCENT: FALLLOW, 2-443
MEASUREMENT: CH<x>: REFLEVELS: PERCENT: FALLMID, 2-444
MEASUREMENT: CH<x>: REFLEVELS: PERCENT: HYSteresis, 2-444
MEASUREMENT: CH<x>: REFLEVELS: PERCENT: RISEHigh, 2-445
MEASUREMENT: CH<x>: REFLEVELS: PERCENT: RISELow, 2-445
MEASUREMENT: CH<x>: REFLEVELS: PERCENT: RISEMID, 2-446
MEASUREMENT: CH<x>: REFLEVELS: PERCENT: TYPE, 2-446
MEASUREMENT: CLOCKRECOVERY: ADVANCED: METHOD, 2-447
MEASUREMENT: CLOCKRECOVERY: CLOCKFREQUENCY, 2-448
MEASUREMENT: CLOCKRECOVERY: CLOCKMULTIPLIER, 2-448
MEASUREMENT: CLOCKRECOVERY: CLOCKRECOVERY: CONSTCLOCKMODE, 2-449
MEASUREMENT: CLOCKRECOVERY: DAMPING, 2-449
MEASUREMENT: CLOCKRECOVERY: DATAPath, 2-450
MEASUREMENT: CLOCKRECOVERY: DATA RATE, 2-450
MEASUREMENT: CLOCKRECOVERY: EXPLICITCLOCKMODE, 2-451
MEASUREMENT: CLOCKRECOVERY: JITTERBANDWIDTH, 2-451
MEASUREMENT: CLOCKRECOVERY: LOOPBANDWIDTH, 2-452
MEASUREMENT: CLOCKRECOVERY: MEANAUTOCALCULATE, 2-452
MEASUREMENT: CLOCKRECOVERY: METHOD, 2-453
MEASUREMENT: CLOCKRECOVERY: MODE, 2-453
MEASUREMENT: CLOCKRECOVERY: NOMINALOFFSET, 2-454
MEASUREMENT: CLOCKRECOVERY: NOMINALOFFSET: SELECTION, 2-454
MEASUREMENT: CLOCKRECOVERY: STANDARD, 2-455
MEASUREMENT: DELETE, 2-455
MEASUREMENT: DIRACMODEL, 2-456
MEASUREMENT: DISPLAYUNITS, 2-456
MEASUREMENT: EDGE<x>, 2-457
MEASUREMENT: EYERENDER, 2-457
MEASUREMENT: FILTERS: BLANKINGTIME, 2-458
MEASUREMENT: FILTERS: HIGHPASS: FREQ, 2-458
MEASUREMENT: FILTERS: HIGHPASS: SPEC, 2-459
MEASUREMENT: FILTERS: LOWPASS: FREQ, 2-459
MEASUREMENT: FILTERS: LOWPASS: SPEC, 2-460
MEASUREMENT: FILTERS: RAMP TIME, 2-460
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: CLOCKRecovery: DATARate</td>
<td>2-489</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: CLOCKRecovery: EXPLICITCLOCKMODE</td>
<td>2-489</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: CLOCKRecovery: GLOBal</td>
<td>2-490</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: CLOCKRecovery: JTFBandwidth</td>
<td>2-491</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: CLOCKRecovery: LOOPBandwidth</td>
<td>2-491</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: CLOCKRecovery: MEANAutoCalculate</td>
<td>2-492</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: CLOCKRecovery: METHOD</td>
<td>2-492</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: CLOCKRecovery: MODel</td>
<td>2-493</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: CLOCKRecovery: NOMINALOFFSET</td>
<td>2-493</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: CLOCKRecovery: NOMINALOFFSET: SELECTIONType</td>
<td>2-494</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: CLOCKRecovery: STANDard</td>
<td>2-494</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: COMMONMode: FILTERS: STATE</td>
<td>2-495</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: COMMONMode: SOURCES</td>
<td>2-495</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: CYCLEmode</td>
<td>2-496</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: DELAY: EDGE&lt;x&gt;</td>
<td>2-496</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: DISPLAYSTAT: ENABLE</td>
<td>2-497</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: EDGE: EDGE&lt;:x&gt;</td>
<td>2-502</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: EDGE: EDGIncre</td>
<td>2-498</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: EDGES: FROMLevel</td>
<td>2-498</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: EDGES: LEVEL</td>
<td>2-499</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: EDGES: LOWERFREQuency</td>
<td>2-499</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: EDGES: N</td>
<td>2-500</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: EDGES: SLEWRATEmethod</td>
<td>2-500</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: EDGES: TOLevel</td>
<td>2-501</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: EDGES: UPPERFREQuency</td>
<td>2-501</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: FILTERS: BLANKingtime</td>
<td>2-502</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: FILTERs GLOBal</td>
<td>2-503</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: FILTERs HIGHPass: FREQ</td>
<td>2-503</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: FILTERs HIGHPass: SPEC</td>
<td>2-504</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: FILTERs LOWPass: FREQ</td>
<td>2-504</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: FILTERs LOWPass: SPEC</td>
<td>2-505</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: FILTERs: RAMPtime</td>
<td>2-505</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: FROMedge</td>
<td>2-506</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: GATING</td>
<td>2-507</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: GATING: ACTive</td>
<td>2-507</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: GATING: GLOBAL</td>
<td>2-508</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: GATING: HYSTEResis</td>
<td>2-509</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: GATING: LOGICSource</td>
<td>2-509</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: GATING: MIDRef</td>
<td>2-510</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: GATING</td>
<td>2-510</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: GATING: SEARCHSource</td>
<td>2-510</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: GATING: GLOBALref</td>
<td>2-510</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: HIGHREFVoltage</td>
<td>2-511</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: IDLETIME</td>
<td>2-512</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: JITTERSummary: DCD</td>
<td>2-512</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: JITTERSummary: DJ, 2-513</td>
<td></td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: JITTERSummary: DJDD</td>
<td>2-513</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: JITTERSummary: DJ, 2-513</td>
<td></td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: JITTERSummary: EYEWIDTHBER</td>
<td>2-514</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: JITTERSummary: NPJ</td>
<td>2-514</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: JITTERSummary: PJ</td>
<td>2-515</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: JITTERSummary: RJDD</td>
<td>2-515</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: JITTERSummary: TIE</td>
<td>2-516</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: JITTERSummary: TIBER</td>
<td>2-516</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: LABel</td>
<td>2-517</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: LOWREFVoltage</td>
<td>2-517</td>
</tr>
<tr>
<td>MEASUREMENT: MEAS&lt;x&gt;: MAXCycle</td>
<td>2-517</td>
</tr>
</tbody>
</table>
MEASUrement MEAS<x> MEASRange:  
  GLOBal, 2-518
MEASUrement MEAS<x> MEASRange:  
  MAX, 2-519
MEASUrement MEAS<x> MEASRange:  
  MIN, 2-519
MEASUrement MEAS<x> MEASRange:  
  STATE, 2-520
MEASUrement MEAS<x> MEASRange:  
  MINCycle, 2-520
MEASUrement MEAS<x>:  
  PATTERNDETECTION, 2-521
MEASUrement MEAS<x>:  
  PATTERNLENght, 2-521
MEASUrement MEAS<x>:  
  PERFREQ EDGE, 2-522
MEASUrement MEAS<x>:  
  POPLarity, 2-523
MEASUrement MEAS<x>:  
  POPLulation:  
    GLOBal, 2-523
MEASUrement MEAS<x>:  
  POPLulation: LIMIT:  
    STATE, 2-524
MEASUrement MEAS<x>:  
  POPLulation: LIMIT:  
    VALue, 2-524
MEASUrement MEAS<x>:  
  REFLevels: ABSolute:  
    FALLHigh, 2-525
MEASUrement MEAS<x>:  
  REFLevels: ABSolute:  
    FALLLow, 2-525
MEASUrement MEAS<x>:  
  REFLevels: ABSolute:  
    FALLMid, 2-526
MEASUrement MEAS<x>:  
  REFLevels: ABSolute:  
    HYSTeresis, 2-527
MEASUrement MEAS<x>:  
  REFLevels: ABSolute:  
    RISEHigh, 2-527
MEASUrement MEAS<x>:  
  REFLevels: ABSolute:  
    RISELow, 2-528
MEASUrement MEAS<x>:  
  REFLevels: ABSolute:  
    RISEMid, 2-529
MEASUrement MEAS<x>:  
  REFLevels: ABSolute:  
    TYPE, 2-530
MEASUrement MEAS<x>:  
  REFLevels: PERCent:  
    FALLHigh, 2-530
MEASUrement MEAS<x>:  
  REFLevels: PERCent:  
    FALLLow, 2-531
MEASUrement MEAS<x>:  
  REFLevels: PERCent:  
    FALLMid, 2-531
MEASUrement MEAS<x>:  
  REFLevels: PERCent:  
    HYSTeresis, 2-532
MEASUrement MEAS<x>:  
  REFLevels: PERCent:  
    RISEHigh, 2-533
MEASUrement MEAS<x>:  
  REFLevels: PERCent:  
    RISELow, 2-533
MEASUrement MEAS<x>:  
  REFLevels: PERCent:  
    RISEMid, 2-534
MEASUrement MEAS<x>:  
  REFLevels: PERCent:  
    TYPE, 2-534
MEASUrement MEAS<x>:  
  REFMode, 2-535
MEASUrement MEAS<x>:  
  REFVoltage, 2-535
MEASUrement MEAS<x>:  
  RESUItls ALLAcqs:  
    MAXimum?, 2-536
MEASUrement MEAS<x>:  
  RESUItls ALLAcqs:  
    MEAN?, 2-536
MEASUrement MEAS<x>:  
  RESUItls ALLAcqs:  
    MINimum?, 2-537
MEASUrement MEAS<x>:  
  RESUItls ALLAcqs:  
    PK2PK?, 2-537
MEASUrement MEAS<x>:  
  RESUItls ALLAcqs:  
    POPLulation?, 2-538
MEASUrement MEAS<x>:  
  RESUItls ALLAcqs:  
    STDDev?, 2-538
MEASUrement MEAS<x>:  
  RESUItls CURRentacq:  
    MAXimum?, 2-539
MEASUrement MEAS<x>:  
  RESUItls CURRentacq:  
    MEAN?, 2-539
MEASUrement MEAS<x>:  
  RESUItls CURRentacq:  
    MINimum?, 2-539
MEASUrement MEAS<x>:  
  RESUItls CURRentacq:  
    PK2PK?, 2-540
MEASUrement MEAS<x>:  
  RESUItls CURRentacq:  
    POPLulation?, 2-540
MEASUrement MEAS<x>:  
  RESUItls CURRentacq:  
    STDDev?, 2-541
MEASUrement MEAS<x>:  
  SIGNALType, 2-541
MEASUrement MEAS<x>:  
  SOURce<x>, 2-542
MEASUrement MEAS<x>:  
  SSC:NOMinal freq, 2-542
MEASUrement MEAS<x>:  
  SSC:NOMinal freq:  
    SELECTIONtype, 2-542
MEASUrement MEAS<x>:  
  TCKAVG, 2-543
MEASUrement MEAS<x>:  
  TIMINGMode, 2-543
MEASUrement MEAS<x>:  
  TOEEdge, 2-544
MEASUrement MEAS<x>:  
  TOEDGESEARCHDIRect, 2-545
MEASUrement MEAS<x>:  
  TRANSition, 2-545
MEASUrement: MEAS<x>: TYPE, 2-546
MEASUrement: MEAS<x>: WINDOWLENGTH, 2-553
MEASUrement: MEASRange: MAX, 2-475
MEASUrement: MEASRange: MIN, 2-476
MEASUrement: MEASRange: STATE, 2-476
MEASUrement: MINUI, 2-554
MEASUrement: POPULATION: LIMIT: STATE, 2-554
MEASUrement: POPULATION: LIMIT: VALUE, 2-555
MEASUrement: REF<x>: REFLevels: ABSOLUTE: FALLHigh, 2-566
MEASUrement: REF<x>: REFLevels: ABSOLUTE: FALLLow, 2-566
MEASUrement: REF<x>: REFLevels: ABSOLUTE: FALLMid, 2-567
MEASUrement: REF<x>: REFLevels: ABSOLUTE: HYSTERESIS, 2-567
MEASUrement: REF<x>: REFLevels: ABSOLUTE: RISEHigh, 2-568
MEASUrement: REF<x>: REFLevels: ABSOLUTE: RISELow, 2-568
MEASUrement: REF<x>: REFLevels: ABSOLUTE: RISEMid, 2-569
MEASUrement: REF<x>: REFLevels: ABSOLUTE: TYPE, 2-569
MEASUrement: REF<x>: REFLevels: BASETop, 2-570
MEASUrement: REF<x>: REFLevels: METHOD, 2-570
MEASUrement: REF<x>: REFLevels: PERCENT: FALLHigh, 2-571
MEASUrement: REF<x>: REFLevels: PERCENT: FALLLow, 2-571
MEASUrement: REF<x>: REFLevels: PERCENT: FALLMid, 2-572
MEASUrement: REF<x>: REFLevels: PERCENT: HYSTERESIS, 2-572
MEASUrement: REF<x>: REFLevels: PERCENT: RISEHigh, 2-573
MEASUrement: REF<x>: REFLevels: PERCENT: RISELow, 2-573
MEASUrement: REF<x>: REFLevels: PERCENT: RISEMid, 2-574
MEASUrement: REF<x>: REFLevels: PERCENT: TYPE, 2-574
MEASUrement: REFLevels: ABSOLUTE: FALLHigh, 2-555
MEASUrement: REFLevels: ABSOLUTE: FALLLow, 2-556
MEASUrement: REFLevels: ABSOLUTE: FALLMid, 2-556
MEASUrement: REFLevels: ABSOLUTE: HYSTERESIS, 2-557
MEASUrement: REFLevels: ABSOLUTE: RISEHigh, 2-557
MEASUrement: REFLevels: ABSOLUTE: RISELow, 2-558
MEASUrement: REFLevels: ABSOLUTE: RISEMid, 2-558
MEASUrement: REFLevels: ABSOLUTE: TYPE, 2-558
MEASUrement: REFLevels: BASETop, 2-559
MEASUrement: REFLevels: JITTERMODE, 2-559
MEASUrement: REFLevels: METHOD, 2-560
MEASUrement: REFLevels: MODE, 2-561
MEASUrement: REFLevels: PERCENT: FALLHigh, 2-561
MEASUrement: REFLevels: PERCENT: FALLLow, 2-562
MEASUrement: REFLevels: PERCENT: FALLMid, 2-562
MEASUrement: REFLevels: PERCENT: HYSTERESIS, 2-563
MEASUrement: REFLevels: PERCENT: RISEHigh, 2-563
MEASUrement: REFLevels: PERCENT: RISELow, 2-564
MEASUrement: REFLevels: PERCENT: RISEMid, 2-564
MEASUrement: REFLevels: PERCENT: TYPE, 2-565
MEASUrement: REFLevels: TYPE, 2-565
MEASUrement: STATISTICS: CYCLEMODE, 2-575

N
NEWpass, 2-576

O
*OPC, 2-576
*OPT?, 2-577

P
PASSWord, 2-577
PAUSE, 2-578
PLOT: ADDNew, 2-578
PLOT: DELete, 2-579
PLOT: LIST?, 2-579
Index

POWer:POWer<x>:ADDNew, 2-588
POWer:POWer<x>:DELete, 2-589
POWer:POWer<x>:AUTOSet, 2-589
POWer:POWer<x>:CLRESPOnse:
  AMP[1-10]Val, 2-589
POWer:POWer<x>:CLRESPOnse:
  AMPMode, 2-590
POWer:POWer<x>:CLRESPOnse:
  CONSTamplitude, 2-591
POWer:POWer<x>:CLRESPOnse:
  FREQ[1-11]Val, 2-591
POWer:POWer<x>:CLRESPOnse:GENerator, 2-592
POWer:POWer<x>:CLRESPOnse:
  IMPEDance, 2-593
POWer:POWer<x>:CLRESPOnse:
  INPUTSOurce, 2-593
POWer:POWer<x>:CLRESPOnse:
  OUTPUTSOurce, 2-594
POWer:POWer<x>:CLRESPOnse:PPD, 2-594
POWer:POWer<x>:CLRESPOnse:
  STARTFREQuency, 2-595
POWer:POWer<x>:CLRESPOnse:
  STOPFREQuency, 2-595
POWer:POWer<x>:CYCLEAmp:
  INPUTSOurce, 2-596
POWer:POWer<x>:CYCLEBase:
  INPUTSOurce, 2-597
POWer:POWer<x>:CYCLEMAX:
  INPUTSOurce, 2-597
POWer:POWer<x>:CYCLEMin:
  INPUTSOurce, 2-598
POWer:POWer<x>:CYCLEPKPK:
  INPUTSOurce, 2-598
POWer:POWer<x>:CYCLETop:
  INPUTSOurce, 2-599
POWer:POWer<x>:DIDT:INPUTSOurce, 2-600
POWer:POWer<x>:DIDT:
  SOURCEEDGEType, 2-600
POWer:POWer<x>:DVDT:INPUTSOurce, 2-601
POWer:POWer<x>:DVDT:
  SOURCEEDGEType, 2-601
POWer:POWer<x>:EFFICIENCY:INPUTType, 2-602
POWer:POWer<x>:EFFICIENCY:
  IOUT1SOUrce, 2-602
POWer:POWer<x>:EFFICIENCY:
  IOUT2SOUrce, 2-603
POWer:POWer<x>:EFFICIENCY:
  IOUT3SOUrce, 2-604
POWer:POWer<x>:EFFICIENCY:
  IOUT4SOUrce, 2-605
POWer:POWer<x>:EFFICIENCY:
  NUMOFOutputs, 2-605
POWer:POWer<x>:EFFICIENCY:
  OUTPUT1Type, 2-606
POWer:POWer<x>:EFFICIENCY:
  OUTPUT2Type, 2-607
POWer:POWer<x>:EFFICIENCY:
  OUTPUT3Type, 2-607
POWer:POWer<x>:EFFICIENCY:
  OUTPUT4Type, 2-608
POWer:POWer<x>:EFFICIENCY:
  VOUT1SOUrce, 2-609
POWer:POWer<x>:EFFICIENCY:
  VOUT2SOUrce, 2-609
POWer:POWer<x>:EFFICIENCY:
  VOUT3SOUrce, 2-610
POWer:POWer<x>:EFFICIENCY:
  VOUT4SOUrce, 2-611
POWer:POWer<x>:EFFICIENCY:VSOUrce, 2-611
POWer:POWer<x>:FREQuency:EDGE, 2-612
POWer:POWer<x>:FREQuency:
  INPUTSOurce, 2-612
POWer:POWer<x>:GATing, 2-613
POWer:POWer<x>:GATing:GLOBal, 2-614
POWer:POWer<x>:HARMONICS:CLASs, 2-614
POWer:POWer<x>:HARMONICS:CMETHod, 2-615
POWer:POWer<x>:HARMONICS:
  FUNDCURRENT, 2-615
POWer:POWer<x>:HARMONICS:ORDER, 2-616

PLOT:PLOT<x>:BATHtub:BER, 2-580
PLOT:PLOT<x>:BATHtub:XAXISUnits, 2-580
PLOT:PLOT<x>:BITType, 2-580
PLOT:PLOT<x>:LABel:COLor, 2-581
PLOT:PLOT<x>:LABel:FONT:BOLD, 2-581
PLOT:PLOT<x>:LABel:FONT:ITALic, 2-582
PLOT:PLOT<x>:LABel:FONT:SIZE, 2-582
PLOT:PLOT<x>:LABel:FONT:TYPE, 2-583
PLOT:PLOT<x>:LABel:FONT:UNDERline, 2-583
PLOT:PLOT<x>:LABel:NAMe, 2-584
PLOT:PLOT<x>:LABel:XPOS, 2-584
PLOT:PLOT<x>:LABel:YPOS, 2-585
PLOT:PLOT<x>:MASK?, 2-585
PLOT:PLOT<x>:NUMBins, 2-585
PLOT:PLOT<x>:SOUrce<x>, 2-586
PLOT:PLOT<x>:SPECtrum:BASE, 2-586
PLOT:PLOT<x>:SPECtrum:DYNRange, 2-587
PLOT:PLOT<x>:TYPE, 2-587

Index-14  MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer
INDEX

POWer.POWER<x>: HARMONICS.HSOURce, 2-616
POWer.POWER<x>: HARMONICS.IPOWER, 2-617
POWer.POWER<x>: HARMONICS.ISOURce, 2-617
POWer.POWER<x>: HARMONICS: LINEFREQuency, 2-618
POWer.POWER<x>: HARMONICS: ODDEven, 2-619
POWer.POWER<x>: HARMONICS: PFActor, 2-619
POWer.POWER<x>: HARMONICS: POWERRating, 2-620
POWer.POWER<x>: HARMONICS: RCURRENT, 2-620
POWer.POWER<x>: HARMONICS: STANDARD, 2-621
POWer.POWER<x>: HARMONICS: STARTFREQuency, 2-622
POWer.POWER<x>: HARMONICS: UNITS, 2-622
POWer.POWER<x>: HARMONICS: VSOURce, 2-623
POWer.POWER<x>: INDUCTANCE: EDGESource, 2-623
POWer.POWER<x>: INDUCTANCE: ISOURce, 2-624
POWer.POWER<x>: INDUCTANCE: VSOURce, 2-625
POWer.POWER<x>: INPUTCAP: ISOURce, 2-625
POWer.POWER<x>: INPUTCAP: PEAKCURRent, 2-626
POWer.POWER<x>: INPUTCAP: PEAKVOLTage, 2-627
POWer.POWER<x>: INPUTCAP: VSOURce, 2-627
POWer.POWER<x>: INRUSH current: INPUTSOurce, 2-628
POWer.POWER<x>: INRUSH current: PEAKCURRent, 2-629
POWer.POWER<x>: IVSINTEGRALV: ISOURce, 2-629
POWer.POWER<x>: IVSINTEGRALV: VSOURce, 2-629
POWer.POWER<x>: LABel, 2-630
POWer.POWER<x>: LINERIPPLE: INPUTSOurce, 2-631
POWer.POWER<x>: LINERIPPLE: LFREQuency, 2-632
POWer.POWER<x>: MAGNETICLOSS: ISOURce, 2-633
POWer.POWER<x>: MAGNETICLOSS: VSOURce, 2-633
POWer.POWER<x>: MAGPROPERTY: AREAofcrosssection, 2-634
POWer.POWER<x>: MAGPROPERTY: EDGESOURce, 2-634
POWer.POWER<x>: MAGPROPERTY: ISOURce, 2-635
POWer.POWER<x>: MAGPROPERTY: LENgth, 2-636
POWer.POWER<x>: MAGPROPERTY: PRIMARYTURNs, 2-636
POWer.POWER<x>: MAGPROPERTY: SEC1SOURce, 2-637
POWer.POWER<x>: MAGPROPERTY: SEC1TURNs, 2-638
POWer.POWER<x>: MAGPROPERTY: SEC2SOURce, 2-639
POWer.POWER<x>: MAGPROPERTY: SEC2TURNs, 2-640
POWer.POWER<x>: MAGPROPERTY: SEC3SOURce, 2-640
POWer.POWER<x>: MAGPROPERTY: SEC3TURNs, 2-641
POWer.POWER<x>: MAGPROPERTY: SEC4SOURce, 2-642
POWer.POWER<x>: MAGPROPERTY: SEC4TURNs, 2-643
POWer.POWER<x>: MAGPROPERTY: SEC5SOURce, 2-644
POWer.POWER<x>: MAGPROPERTY: SEC5TURNs, 2-645
POWer.POWER<x>: MAGPROPERTY: SEC6SOURce, 2-645
POWer.POWER<x>: MAGPROPERTY: SEC6TURNs, 2-646
POWer.POWER<x>: MAGPROPERTY: SECWINDings, 2-647
POWer.POWER<x>: MAGPROPERTY: UNITS, 2-648
POWer.POWER<x>: MAGPROPERTY: VSOURce, 2-649
POWer.POWER<x>: NDUTYCYCLE: EDGEType, 2-650
POWer.POWER<x>: NDUTYCYCLE: INPUTSOurce, 2-650
POWer.POWER<x>: NPULSEWIDTH: INPUTSOurce, 2-651
POWer.POWER<x>: PDUTYCYCLE: EDGEType, 2-651
POWer.POWER<x>: PDUTYCYCLE: INPUTSOurce, 2-652
POWer.POWER<x>: PERIOD: EDGE, 2-652
POWer.POWER<x>: PERIOD: INPUTSOurce, 2-653
Index-16 MSO54, MSO56, MSO58, MSO58LP, MSO64 Programmer

POWeR:POWeRx> POWERQUALITY: CCYCles, 2-653
POWeR:POWeRx> POWERQUALITY: FREFerence, 2-654
POWeR:POWeRx> POWERQUALITY: ISOURce, 2-654
POWeR:POWeRx> POWERQUALITY: VSOURce, 2-655
POWeR:POWeRx> PPULSEWIDTH: INPUTSO urce, 2-655
POWeR:POWeRx> PRESET, 2-656
POWeR:POWeRx> PSRR:AMP[1-10]Val, 2-656
POWeR:POWeRx> PSRR: AMPMode, 2-657
POWeR:POWeRx> PSRR: CONSTAMPlitude, 2-658
POWeR:POWeRx> PSRR: GENERator, 2-659
POWeR:POWeRx> PSRR: IMPEDance, 2-660
POWeR:POWeRx> PSRR: INPUTSOurce, 2-660
POWeR:POWeRx> PSRR: OUTPUTSOurce, 2-661
POWeR:POWeRx> PSRR: PPD, 2-661
POWeR:POWeRx> PSRR: STARTFREQuency, 2-662
POWeR:POWeRx> PSRR: STOPFREQuency, 2-662
POWeR:POWeRx> RDSON: DEVICEType, 2-663
POWeR:POWeRx> RDSON: ISOURce, 2-664
POWeR:POWeRx> RDSON: VSOURce, 2-664
POWeR:POWeRx> REFLevels ABSolute: FALLHigh, 2-665
POWeR:POWeRx> REFLevels ABSolute: FALLLow, 2-666
POWeR:POWeRx> REFLevels ABSolute: FALLMid, 2-666
POWeR:POWeRx> REFLevels ABSolute: HYSTeresis, 2-667
POWeR:POWeRx> REFLevels ABSolute: RISEHigh, 2-667
POWeR:POWeRx> REFLevels ABSolute: RISELow, 2-668
POWeR:POWeRx> REFLevels ABSolute: RISEMid, 2-669
POWeR:POWeRx> REFLevels ABSolute: TYPE, 2-669
POWeR:POWeRx> REFLevels BASETop, 2-670
POWeR:POWeRx> REFLevels METHOD, 2-670
POWeR:POWeRx> REFLevels PERCent: FALLHigh, 2-671
POWeR:POWeRx> REFLevels PERCent: FALLLow, 2-671
POWeR:POWeRx> REFLevels PERCent: FALLMid, 2-672
POWeR:POWeRx> REFLevels PERCent: HYSTeresis, 2-673
POWeR:POWeRx> REFLevels PERCent: RISEHigh, 2-673
POWeR:POWeRx> REFLevels PERCent: RISELow, 2-674
POWeR:POWeRx> REFLevels PERCent: RISEMid, 2-674
POWeR:POWeRx> REFLevels PERCent: TYPE, 2-675
POWeR:POWeRx> RESUItls ALLA.cqs: MAXimum, 2-676
POWeR:POWeRx> RESUItls ALLA.cqs: MEAN?, 2-678
POWeR:POWeRx> RESUItls ALLA.cqs: MINimum?, 2-680
POWeR:POWeRx> RESUItls ALLA.cqs: PK2PK?, 2-682
POWeR:POWeRx> RESUItls ALLA.cqs: POPUlation?, 2-685
POWeR:POWeRx> RESUItls ALLA.cqs: STDDev?, 2-687
POWeR:POWeRx> RESUItls CURRentacq: F1MAG?, 2-689
POWeR:POWeRx> RESUItls CURRentacq: F3MAG?, 2-689
POWeR:POWeRx> RESUItls CURRentacq: FREQUENCY?, 2-690
POWeR:POWeRx> RESUItls CURRentacq: IRMS?, 2-690
POWeR:POWeRx> RESUItls CURRentacq: MAXimum?, 2-691
POWeR:POWeRx> RESUItls CURRentacq: MEAN?, 2-693
POWeR:POWeRx> RESUItls CURRentacq: MINimum?, 2-695
POWeR:POWeRx> RESUItls CURRentacq: PK2PK?, 2-697
POWeR:POWeRx> RESUItls CURRentacq: POHCL?, 2-699
POWeR:POWeRx> RESUItls CURRentacq: POHCM?, 2-700
POWeR:POWeRx> RESUItls CURRentacq: POHCS?, 2-700
POWer:POWer<x>:RESUlts:CURRentacq:POPUlation?, 2-700
POWer:POWer<x>:RESUlts:CURRentacq:RMS?, 2-703
POWer:POWer<x>:RESUlts:CURRentacq:STATUS?, 2-703
POWer:POWer<x>:RESUlts:CURRentacq:STDDev?, 2-704
POWer:POWer<x>:RESUlts:CURRentacq:THDF?, 2-706
POWer:POWer<x>:RESUlts:CURRentacq:THDR?, 2-706
POWer:POWer<x>:RESUlts:CURRentacq:TRPWR?, 2-707
POWer:POWer<x>:RESUlts:CURRentacq:VRMS?, 2-707
POWer:POWer<x>:SEQSETup, 2-708
POWer:POWer<x>:SEQUence, 2-708
POWer:POWer<x>:SOA:ISOURce, 2-709
POWer:POWer<x>:SOA:POINT, 2-709
POWer:POWer<x>:SOA:RECALLmask, 2-710
POWer:POWer<x>:SOA:RECALLmask:FILENAME, 2-710
POWer:POWer<x>:SOA:SAVemask, 2-711
POWer:POWer<x>:SOA:SAVemask:AUTOINCrement, 2-711
POWer:POWer<x>:SOA:SAVemask:FILENAME, 2-712
POWer:POWer<x>:SOA:SAVemask:FILEDName, 2-712
POWer:POWer<x>:SOA:SAVemask:FILEDName, 2-712
POWer:POWer<x>:SWITCHINGLOSS:DEVICEType, 2-713
POWer:POWer<x>:SWITCHINGLOSS:GATESOurce, 2-714
POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts, 2-714
POWer:POWer<x>:SWITCHINGLOSS:RDSOn, 2-715
POWer:POWer<x>:SWITCHINGLOSS:SWLCONFIgType, 2-717
POWer:POWer<x>:SWITCHINGLOSS:VCESat, 2-718
POWer:POWer<x>:SWITCHINGLOSS:VLevel, 2-719
POWer:POWer<x>:SWITCHINGLOSS:VLEVELAbts, 2-719
POWer:POWer<x>:SWITCHINGLOSS:VLEVELPct, 2-720
POWer:POWer<x>:SWITCHINGLOSS:VSOURce, 2-720
POWer:POWer<x>:SWITCHINGGRIPPLE:INPUTSOurce, 2-721
POWer:POWer<x>:SWITCHINGGRIPPLE:LFREQuency, 2-722
POWer:POWer<x>:SWITCHINGGRIPPLE:MAXTIMe, 2-724
POWer:POWer<x>:SWITCHINGGRIPPLE:MAXV oltage, 2-725
POWer:POWer<x>:SWITCHINGGRIPPLE:NUMOUTputs, 2-725
POWer:POWer<x>:SWITCHINGGRIPPLE:OUTPUT1SOURce, 2-727
POWer:POWer<x>:SWITCHINGGRIPPLE:OUTPUT1VOLTage, 2-727
POWer:POWer<x>:SWITCHINGGRIPPLE:OUTPUT2SOURce, 2-728
POWer:POWer<x>:SWITCHINGGRIPPLE:OUTPUT2VOLTage, 2-728
POWer:POWer<x>:SWITCHINGGRIPPLE:OUTPUT3SOURce, 2-729
POWer:POWer<x>:SWITCHINGGRIPPLE:OUTPUT3VOLTage, 2-729
POWer:POWer<x>:SWITCHINGGRIPPLE:OUTPUT3SOURce, 2-729
POWer:POWer<x>:SWITCHINGGRIPPLE:OUTPUT3VOLTage, 2-730
POWer:POWer<x>:SWITCHINGGRIPPLE:OUTPUT4SOURce, 2-731
POWer:POWer<x>:SWITCHINGGRIPPLE:OUTPUT4VOLTage, 2-731
POWer:POWer<x>:SWITCHINGGRIPPLE:OUTPUT5SOURce, 2-732
POWer:POWer<x>:SWITCHINGGRIPPLE:OUTPUT5VOLTage, 2-733
POWer:POWer<x>:TURNOFFtime:
  OUTPUT6SOURce, 2-734
POWer:POWer<x>:TURNOFFtime:
  OUTPUT6VOLTage, 2-734
POWer:POWer<x>:TURNOFFtime:
  OUTPUT7SOURce, 2-735
POWer:POWer<x>:TURNOFFtime:
  OUTPUT7VOLTage, 2-736
POWer:POWer<x>:TURNOFFtime:TYPE, 2-736
POWer:POWer<x>:TURNONtime:
  FREQuency, 2-737
POWer:POWer<x>:TURNONtime:
  INPUTLEVEL, 2-738
POWer:POWer<x>:TURNONtime:
  INPUTSOURce, 2-738
POWer:POWer<x>:TURNONtime:
  MAXTIMe, 2-739
POWer:POWer<x>:TURNONtime:
  MAXVoltage, 2-739
POWer:POWer<x>:TURNONtime:
  NUMOUTputs, 2-740
POWer:POWer<x>:TURNONtime:
  OUTPUT1SOURce, 2-741
POWer:POWer<x>:TURNONtime:
  OUTPUT1VOLTage, 2-742
POWer:POWer<x>:TURNONtime:
  OUTPUT2SOURce, 2-743
POWer:POWer<x>:TURNONtime:
  OUTPUT2VOLTage, 2-743
POWer:POWer<x>:TURNONtime:
  OUTPUT3SOURce, 2-744
POWer:POWer<x>:TURNONtime:
  OUTPUT3VOLTage, 2-745
POWer:POWer<x>:TURNONtime:
  OUTPUT4SOURce, 2-745
POWer:POWer<x>:TURNONtime:
  OUTPUT4VOLTage, 2-746
POWer:POWer<x>:TURNONtime:
  OUTPUT5SOURce, 2-747
POWer:POWer<x>:TURNONtime:
  OUTPUT5VOLTage, 2-747
POWer:POWer<x>:TURNONtime:
  OUTPUT6SOURce, 2-748
POWer:POWer<x>:TURNONtime:
  OUTPUT6VOLTage, 2-749
POWer:POWer<x>:TURNONtime:
  OUTPUT7SOURce, 2-750
POWer:POWer<x>:TURNONtime:
  OUTPUT7VOLTage, 2-750
POWer:POWer<x>:TURNONtime:
  TYPE, 2-751
POWer:POWer<x>:TURNONtime:
  ADDNew, 2-752
POWer:POWer<x>:TURNONtime:
  DELete, 2-753
POWer:POWer<x>:TURNONtime:
  LIST?, 2-753
*PSC, 2-753
*PUD, 2-754

R
RECALL:SESSion, 2-755
RECALL:SEtUP, 2-756
RECALL:WA VEform, 2-756
REF:ADDNew, 2-757
REF:DELete, 2-757
REF:LIST?, 2-758
REF:REF<x>:DESKew, 2-758
REF:REF<x>:LABel:COLor, 2-759
REF:REF<x>:LABel:FONT:BOLD, 2-759
REF:REF<x>:LABel:FONT:ITALic, 2-759
REF:REF<x>:LABel:FONT:SIZE, 2-760
REF:REF<x>:LABel:FONT:TYPE, 2-760
REF:REF<x>:LABel:UNDERline, 2-761
REF:REF<x>:LABel:NAMe, 2-761
REF:REF<x>:LABel:XPOS, 2-762
REF:REF<x>:LABel:YPOS, 2-762
REF:REF<x>:SOUrce, 2-763
REF:<D:LABel:COLor, 2-767
REF:<D:LABel:FONT:BOLD, 2-768
REF:<D:LABel:FONT:ITALic, 2-768
REF:<D:LABel:FONT:SIZE, 2-769
REF:<D:LABel:UNDERline, 2-770
REF:<D:LABel:NAMe, 2-770
REF:<D:LABel:XPOS, 2-770
REF:<D:LABel:YPOS, 2-771
REF:<DALL:LABel:COLor, 2-763
REF:<DALL:LABel:FONT:BOLD, 2-764
REF:<DALL:LABel:FONT:ITALic, 2-764
REF:<DALL:LABel:FONT:SIZE, 2-765
REF:<DALL:LABel:UNDERline, 2-766
REF:<DALL:LABel:NAMe, 2-766
REF:<DALL:LABel:XPOS, 2-766
REF:<DALL:LABel:YPOS, 2-767
REM, 2-771
Index

SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:CONDition, 2-809
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:DATa:HIVALue, 2-810
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:DATa:OFFSET, 2-811
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:DATa:QUALifier, 2-812
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:DATa:SIZE, 2-813
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:DATa:VALUE, 2-813
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:IPHeader:DESTinationaddr:VALUE, 2-814
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:IPHeader:PROTOCOL:VALUE, 2-815
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:IPHeader:SOURcaddr:VALUE, 2-816
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:MAC:ADDRess:DESTination:VALUE, 2-816
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:MAC:ADDRess:SOURce:VALUE, 2-817
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:MAC:LENGTH:HIVALue, 2-818
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:MAC:LENGTH:VALUE, 2-819
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:QTAG:VALUE, 2-819
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:TCPHeader:ACKnum:VALUE, 2-820
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:TCPHeader:DESTinationport:VALUE, 2-821
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:TCPHeader:SEQUnce:VALUE, 2-822
SEARCH:SEARCH<x>:TRIGger:A:BUS:
ETHERnet:TCPHeader:SOURceport:VALUE, 2-822
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:CONDition, 2-833
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:CYCLEcount:HIVALue, 2-824
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:CYCLEcount:VALUE, 2-825
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:DATa:HIVALue, 2-826
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:DATa:OFFSET, 2-826
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:DATa:QUALifier, 2-827
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:DATa:SIZE, 2-828
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:DATa:VALUE, 2-828
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:EOFTYPE, 2-829
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:ERRTYPE, 2-829
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXray:FRAMEID:HIVALue, 2-830
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXray:FRAMEID:QUALifier, 2-830
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXray:FRAMEID:VALUE, 2-831
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXray:FRAMEType, 2-831
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:HEADER:CRC, 2-832
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:HEADER:CYCLECount, 2-833
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:HEADER:FRAMEID, 2-833
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:HEADER:INDBits, 2-834
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:HEADER:PAYLengh, 2-834
SEARCH:SEARCH<x>:TRIGger:A:BUS:12C:ADDRESS MODE, 2-835
SEARCH:SEARCH<x>:TRIGger:A:BUS:12C:ADDRESS:VALUE, 2-836
SEARCH:SEARCH<x>:TRIGger:A:BUS:12C:CONDition, 2-836
SEARCH:SEARCH<x>:TRIGger:A:BUS:12C:DIRECTion, 2-837
SEARCH:SEARCH<x>:TRIGger:A:BUS:12C:DATA:SIZE, 2-838
SEARCH:SEARCH<x>:TRIGger:A:BUS:12C:DATA:VALUE, 2-838
SEARCH. SEARCH<>: TRIGGER A: BUS: SENT. 
  FAST: COUNTer: VALUE, 2-869
SEARCH. SEARCH<>: TRIGGER A: BUS: SENT. 
  FAST: INVERTNIBLE: VALUE, 2-869
SEARCH. SEARCH<>: TRIGGER A: BUS: SENT. 
  FAST: STATUS: VALUE, 2-870
SEARCH. SEARCH<>: TRIGGER A: BUS: SENT. 
  PAUSE: QUALifier, 2-871
SEARCH. SEARCH<>: TRIGGER A: BUS: SENT. 
  PAUSE: TICKs: VALUE, 2-872
SEARCH. SEARCH<>: TRIGGER A: BUS: SENT. 
  PAUSE: TICKs: VALUE, 2-873
SEARCH. SEARCH<>: TRIGGER A: BUS: SENT. 
  SLOW: DATA: VALUE, 2-873
SEARCH. SEARCH<>: TRIGGER A: BUS: SENT. 
  SLOW: DATA: VALUE, 2-874
SEARCH. SEARCH<>: TRIGGER A: BUS: SENT. 
  SLOW: DATA: VALUE, 2-875
SEARCH. SEARCH<>: TRIGGER A: BUS: SENT. 
  SLOW: IDENTIFIER: VALUE, 2-876
SEARCH. SEARCH<>: TRIGGER A: BUS: SOURCE, 2-876
SEARCH. SEARCH<>: TRIGGER A: BUS: SPI: 
  CONDITION, 2-877
SEARCH. SEARCH<>: TRIGGER A: BUS: SPI: 
  DATA: SIZE, 2-877
SEARCH. SEARCH<>: TRIGGER A: BUS: SPI: DATA: 
  VALUE, 2-878
SEARCH. SEARCH<>: TRIGGER A: BUS: SPI: 
  CONDITION, 2-879
SEARCH. SEARCH<>: TRIGGER A: USB: SPI: DATA: SIZE, 2-880
SEARCH. SEARCH<>: TRIGGER A: USB: SPI: 
  MASTER ADDRESS: VALUE, 2-881
SEARCH. SEARCH<>: TRIGGER A: USB: SPI: 
  NO RESPONSE, 2-882
SEARCH. SEARCH<>: TRIGGER A: USB: SPI: 
  REGISTER ADDRESS: VALUE, 2-883
SEARCH. SEARCH<>: TRIGGER A: USB: SPI: 
  SLAVE ADDRESS: VALUE, 2-883
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  ADDRESS: VALUE, 2-884
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  ADDRESS: VALUE, 2-885
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  CONDITION, 2-886
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  DATA: VALUE, 2-886
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  DATA: OFFSET, 2-887
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  DATA: QUALIFIER, 2-888
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  DATA: TYPE, 2-889
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  DATA: VALUE, 2-890
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  END POINT: VALUE, 2-890
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  SOFFRAMENUMBER, 2-891
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  SPECIAL TYPE, 2-893
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  SPLIT: ET: VALUE, 2-894
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  SPLIT: HUB: VALUE, 2-894
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  SPLIT: PORT: VALUE, 2-895
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  SPLIT: SC: VALUE, 2-896
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  SPLIT: SE: VALUE, 2-896
SEARCH. SEARCH<>: TRIGGER A: USB: USB: 
  TOKEN TYPE, 2-897
SEARCH. SEARCH<>: TRIGGER A: DDRREAD: 
  DATARATE, 2-898
SEARCH. SEARCH<>: TRIGGER A: DDRREAD: 
  DATA SOURCE, 2-899
SEARCH. SEARCH<>: TRIGGER A: DDRREAD: 
  HYSTERESIS, 2-899
SEARCH. SEARCH<>: TRIGGER A: DDRREAD: 
  марг, 2-900
SEARCH. SEARCH<>: TRIGGER A: DDRREAD: 
  POSTAMBLE LENGTH, 2-901
<table>
<thead>
<tr>
<th>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TIMEOut:</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME, 2-956</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>DELTA Time, 2-957</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>LOGIC QUALification, 2-957</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>POL arity, 2-958</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>SOURCE, 2-958</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>THRESHOLD: HIGH, 2-959</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>THRESHOLD: LOW, 2-959</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>WHEN, 2-960</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>TYPE, 2-960</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>WINDOW: CROSSING, 2-962</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>WINDOW: LOGIC QUALIFICATION, 2-963</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>POL arity, 2-963</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>SOURCE, 2-964</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>THRESHOLD: HIGH, 2-965</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>THRESHOLD: LOW, 2-965</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>WHEN, 2-966</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;: TRIGGER:A: TRANSition:</td>
</tr>
<tr>
<td>WINDOW: WIDTH, 2-966</td>
</tr>
<tr>
<td>SEARCH: SELECTed, 2-967</td>
</tr>
<tr>
<td>SEARCHTABLE, 2-967</td>
</tr>
<tr>
<td>SET?, 2-968</td>
</tr>
<tr>
<td>SOCKET Server ENABLE, 2-969</td>
</tr>
<tr>
<td>SOCKET Server PORT, 2-969</td>
</tr>
<tr>
<td>SOCKET Server PROTOCOL, 2-970</td>
</tr>
<tr>
<td>*SRE, 2-970</td>
</tr>
<tr>
<td>*STB?, 2-971</td>
</tr>
</tbody>
</table>

**T**

TEK Secure, 2-972
TIME?, 2-973
TIME: ZONE, 2-973
TIME: ZONE: UTC DELTa, 2-973

TOTAL uptime?, 2-973
TOUCH SCREEN: CALibrate, 2-974
TOUCH SCREEN: STATE, 2-974
*TRG, 2-975
TRIGGER, 2-975
TRIGGER: (A|B): BUS: <x>: ARINC429A: CONDITION, 2-976
TRIGGER: (A|B): BUS: <x>: ARINC429A: ERR TYPE, 2-979
TRIGGER: (A|B): BUS: <x>: ARINC429A: LABEL: HIGH VALUE, 2-980
TRIGGER: (A|B): BUS: <x>: AUDI o: CONDITION, 2-983
TRIGGER: (A|B): BUS: <x>: AUDI o: DATA: HIGH VALUE, 2-984
TRIGGER: (A|B): BUS: <x>: AUDI o: DATA: OFFSET, 2-985
TRIGGER: (A|B): BUS: <x>: AUDI o: DATA: QUALIFIER, 2-985
TRIGGER: (A|B): BUS: <x>: AUDI o: DATA: VALUE, 2-987
TRIGGER: (A|B): BUS: <x>: CAN CONDITION, 2-988
TRIGGER: (A|B): BUS: <x>: CAN DATA: DIRECTION, 2-989
TRIGGER: (A|B): BUS: <x>: CAN DATA: OFFSET, 2-989
Index

TRIGger.(A|B):BUS.B<x>:CAN:DATA:
QUALifier, 2-990
TRIGger.(A|B):BUS.B<x>:CAN:DATA:SIZE, 2-991
TRIGger.(A|B):BUS.B<x>:CAN:DATA:
VALUE, 2-992
TRIGger.(A|B):BUS.B<x>:CAN:ERRType, 2-992
TRIGger.(A|B):BUS.B<x>:CAN:FD:BRSBit, 2-993
TRIGger.(A|B):BUS.B<x>:CAN:FD:ESIBit, 2-994
TRIGger.(A|B):BUS.B<x>:CAN:
FRAMEtype, 2-994
TRIGger.(A|B):BUS.B<x>:CAN:IDenti:
IFIER, 2-995
TRIGger.(A|B):BUS.B<x>:CAN:IDenti:
VALUE, 2-996
TRIGger.(A|B):BUS.B<x>:CAN:IDenti:
VALUE, 2-997
TRIGger.(A|B):BUS.B<x>:CAN:IDenti:
VALUE, 2-998
TRIGger.(A|B):BUS.B<x>:CAN:IDenti:
SIZE, 2-999
TRIGger.(A|B):BUS.B<x>:CAN:IDenti:
VALUE, 2-1000
TRIGger.(A|B):BUS.B<x>:CAN:IPHeader:
DESTinationaddr:VALUE, 2-1001
TRIGger.(A|B):BUS.B<x>:CAN:IPHeader:
PROTocol:VALUE, 2-1002
TRIGger.(A|B):BUS.B<x>:CAN:IPHeader:
SOURCEaddr:VALUE, 2-1003
TRIGger.(A|B):BUS.B<x>:CAN:MAC:
ADDRESS:DESTinationVALUE, 2-1004
TRIGger.(A|B):BUS.B<x>:CAN:MAC:
ADDRESS:SOURCEVALUE, 2-1005
TRIGger.(A|B):BUS.B<x>:CAN:MAC:
LENGTH:HIVALue, 2-1005
TRIGger.(A|B):BUS.B<x>:CAN:MAC:
LENGTH:VALUE, 2-1006
TRIGger.(A|B):BUS.B<x>:CAN:QTAG:
VALUE, 2-1007
TRIGger.(A|B):BUS.B<x>:CAN:TCPHeader:
ACKnum:VALUE, 2-1008
TRIGger.(A|B):BUS.B<x>:CAN:TCPHeader:
DESTinationport:VALUE, 2-1009
TRIGger.(A|B):BUS.B<x>:ETHERnet:TCPHeader:
SEQnum:VALUE, 2-1009
TRIGger.(A|B):BUS.B<x>:ETHERnet:TCPHeader:
SOURCEport:VALUE, 2-1010
TRIGger.(A|B):BUS.B<x>:FLEXray:CYCLEcount:
HIVALue, 2-1011
TRIGger.(A|B):BUS.B<x>:FLEXray:CYCLEcount:
QUALifier, 2-1012
TRIGger.(A|B):BUS.B<x>:FLEXray:CYCLEcount:
VALUE, 2-1013
TRIGger.(A|B):BUS.B<x>:FLEXray:DATa:
HIVALue, 2-1014
TRIGger.(A|B):BUS.B<x>:FLEXray:DATa:
OFFSET, 2-1015
TRIGger.(A|B):BUS.B<x>:FLEXray:DATa:
QUALifier, 2-1016
TRIGger.(A|B):BUS.B<x>:FLEXray:DATa:
VALUE, 2-1017
TRIGger.(A|B):BUS.B<x>:FLEXray:
EOFTYPE, 2-1018
TRIGger.(A|B):BUS.B<x>:FLEXray:
ERRORTYPE, 2-1019
TRIGger.(A|B):BUS.B<x>:FLEXray:FRAMEID:
HIVALue, 2-1020
TRIGger.(A|B):BUS.B<x>:FLEXray:FRAMEID:
QUALifier, 2-1021
TRIGger.(A|B):BUS.B<x>:FLEXray:FRAMEID:
VALUE, 2-1022
TRIGger.(A|B):BUS.B<x>:FLEXray:
FRAMEType, 2-1023
TRIGger.(A|B):BUS.B<x>:FLEXray:HEADER:
CRC, 2-1024
TRIGger.(A|B):BUS.B<x>:FLEXray:HEADER:
CYCLEcount, 2-1025
TRIGger.(A|B):BUS.B<x>:FLEXray:HEADER:
FRAMEID, 2-1026
TRIGger.(A|B):BUS.B<x>:FLEXray:HEADER:
INDBits, 2-1027
TRIGger.(A|B):BUS.B<x>:FLEXray:HEADER:
PAYLength, 2-1028
TRIGger.(A|B):BUS.B<x>:I2C:
ADDRESS:
MODE, 2-1029
TRIGGER {A|B} : BUS <x> I2C ADDRESS: VALUE, 2-1025
TRIGGER {A|B} : BUS <x> I2C CONDITION, 2-1026
TRIGGER {A|B} : BUS <x> I2C DATA: DIRECT, 2-1027
TRIGGER {A|B} : BUS <x> I2C DATA: SIZE, 2-1027
TRIGGER {A|B} : BUS <x> I2C DATA: VALUE, 2-1028
TRIGGER {A|B} : BUS <x> LIN CONDITION, 2-1029
TRIGGER {A|B} : BUS <x> LIN DATA: QUALIFIER, 2-1030
TRIGGER {A|B} : BUS <x> LIN DATA: VALUE, 2-1031
TRIGGER {A|B} : BUS <x> LIN ERRTYPE, 2-1032
TRIGGER {A|B} : BUS <x> LIN IDENTIFIER: VALUE, 2-1032
TRIGGER {A|B} : BUS <x> MIL1553B: COMMAND ADDRESS: HIVALUE, 2-1033
TRIGGER {A|B} : BUS <x> MIL1553B: COMMAND ADDRESS: QUALIFIER, 2-1034
TRIGGER {A|B} : BUS <x> MIL1553B: COMMAND ADDRESS: VALUE, 2-1034
TRIGGER {A|B} : BUS <x> MIL1553B: COMMAND COUNT, 2-1035
TRIGGER {A|B} : BUS <x> MIL1553B: COMMAND PARITY, 2-1035
TRIGGER {A|B} : BUS <x> MIL1553B: COMMAND SUB ADDRESS, 2-1036
TRIGGER {A|B} : BUS <x> MIL1553B: COMMAND TRBIT, 2-1037
TRIGGER {A|B} : BUS <x> MIL1553B: CONDITION: PARITY, 2-1038
TRIGGER {A|B} : BUS <x> MIL1553B: DATA: VALUE, 2-1039
TRIGGER {A|B} : BUS <x> MIL1553B: ERRTYPE, 2-1039
TRIGGER {A|B} : BUS <x> MIL1553B: STATUS: ADDRESS: HIVALUE, 2-1040
TRIGGER {A|B} : BUS <x> MIL1553B: STATUS: ADDRESS: QUALIFIER, 2-1040
TRIGGER {A|B} : BUS <x> MIL1553B: STATUS: ADDRESS: VALUE, 2-1041
TRIGGER {A|B} : BUS <x> MIL1553B: STATUS: BIT: BCR, 2-1042
TRIGGER {A|B} : BUS <x> MIL1553B: STATUS: BIT: BUSY, 2-1042
TRIGGER {A|B} : BUS <x> MIL1553B: STATUS: BIT: DBCA, 2-1043
TRIGGER {A|B} : BUS <x> MIL1553B: STATUS: BIT: INSTR, 2-1043
TRIGGER {A|B} : BUS <x> MIL1553B: STATUS: BIT: ME, 2-1044
TRIGGER {A|B} : BUS <x> MIL1553B: STATUS: BIT: SRQ, 2-1045
TRIGGER {A|B} : BUS <x> MIL1553B: STATUS: BIT: SUBSF, 2-1045
TRIGGER {A|B} : BUS <x> MIL1553B: STATUS: BIT: TF, 2-1046
TRIGGER {A|B} : BUS <x> MIL1553B: STATUS: PARITY, 2-1047
TRIGGER {A|B} : BUS <x> MIL1553B: TIME: LESSLIMIT, 2-1047
TRIGGER {A|B} : BUS <x> MIL1553B: TIME: MORELIMIT, 2-1048
TRIGGER {A|B} : BUS <x> MIL1553B: TIME: QUALIFIER, 2-1049
TRIGGER {A|B} : BUS <x> MIL1553B: PARALLEL DATA: VALUE, 2-1049
TRIGGER {A|B} : BUS <x> RS232C: CONDITION, 2-1050
TRIGGER {A|B} : BUS <x> RS232C DATA: SIZE, 2-1050
TRIGGER {A|B} : BUS <x> RS232C DATA: VALUE, 2-1051
TRIGGER {A|B} : BUS <x> SENT: CONDITION, 2-1052
TRIGGER {A|B} : BUS <x> SENT ERRTYPE: CRC, 2-1053
TRIGGER {A|B} : BUS <x> SENT ERRTYPE: 2-1052
TRIGGER {A|B} : BUS <x> SENT FAST CHAN1A: HIVALUE, 2-1054
TRIGGER {A|B} : BUS <x> SENT FAST CHAN1A: QUALIFIER, 2-1054
TRIGGER {A|B} : BUS <x> SENT FAST CHAN1A: VALUE, 2-1055
TRIGGER {A|B} : BUS <x> SENT FAST CHAN2B: HIVALUE, 2-1056
TRIGGER {A|B} : BUS <x> SENT FAST CHAN2B: QUALIFIER, 2-1057
TRIGger (A|B): BUS <x>: SENT: FAST: CHANNEL: B: VALUE, 2-1058
TRIGger (A|B): BUS <x>: SENT: FAST: COUNTER: HIVALue, 2-1058
TRIGger (A|B): BUS <x>: SENT: FAST: COUNTER: VALUE, 2-1060
TRIGger (A|B): BUS <x>: SENT: FAST: INVERT: NIBLE: VALUE, 2-1061
TRIGger (A|B): BUS <x>: SENT: FAST: STATUS: VALUE, 2-1062
TRIGger (A|B): BUS <x>: SENT: PAUSE: QUALifier, 2-1063
TRIGger (A|B): BUS <x>: SENT: SLOW: DATA: HIVALue, 2-1064
TRIGger (A|B): BUS <x>: SENT: SLOW: DATA: QUALifier, 2-1065
TRIGger (A|B): BUS <x>: SENT: SLOW: IDENTIFIER: VALUE, 2-1066
TRIGger (A|B): BUS <x>: SPI: CONDITION, 2-1067
TRIGger (A|B): BUS <x>: SPI: DATA: SIZE, 2-1068
TRIGger (A|B): BUS <x>: SPI: DATA: VALUE, 2-1069
TRIGger (A|B): BUS <x>: SPI: IDENTIFIER: SIZE, 2-1070
TRIGger (A|B): BUS <x>: SPI: MasterADDRESS: VALUE, 2-1071
TRIGger (A|B): BUS <x>: SPI: NORESPonse, 2-1072
TRIGger (A|B): BUS <x>: SPI: REGISTERADDRESS: VALUE, 2-1073
TRIGger (A|B): BUS <x>: SPI: SLAVEADDRESS: HIVALue, 2-1074
TRIGger (A|B): BUS <x>: USB: ADDRESS: VALUE, 2-1075
TRIGger (A|B): BUS <x>: USB: ADDRESS: HIVALue, 2-1076
TRIGger (A|B): BUS <x>: USB: DATA: OFFSET, 2-1077
TRIGger (A|B): BUS <x>: USB: DATA: QUALifier, 2-1078
TRIGger (A|B): BUS <x>: USB: DATA: SIZE, 2-1079
TRIGger (A|B): BUS <x>: USB: DATA: TYPE, 2-1080
TRIGger (A|B): BUS <x>: USB: ENDPoint: VALUE, 2-1081
TRIGger (A|B): BUS <x>: USB: ERRType, 2-1082
TRIGger (A|B): BUS <x>: USB: HANDLEStype, 2-1083
TRIGger (A|B): BUS <x>: USB: MESSAGE: SIZE, 2-1084
TRIGger (A|B): BUS <x>: USB: MESSAGE: STRING, 2-1085
TRIGger (A|B): BUS <x>: USB: MESSAGE: VALUE, 2-1086
TRIGger (A|B): BUS <x>: USB: TOKENType, 2-1087
TRIGger (A|B): BUS: SOUrce, 2-1088
TRIGger (A|B): EDGE: COUPling, 2-1089
TRIGger (A|B): EDGE: SLOpe, 2-1090
TRIGger (A|B): EDGE: SOUrce, 2-1091
TRIGger (A|B): EDGE: USEClEckedge, 2-1092
TRIGger (A|B): EDGE: WHERE, 2-1093
TRIGger (A|B): LOGIc: DELETetime, 2-1094
TRIGger (A|B): LOGIc: FUNCTION, 2-1095
TRIGger (A|B): LOGIc: INPUT: CLOCK: SOUrce, 2-1096
TRIGger (A|B): LOGIc: POLarity, 2-1097
TRIGger (A|B): LOGIc: USEClEckedge, 2-1098
TRIGger (A|B): LOGIc: WHENe, 2-1099
TRIGger (A|B): LOGIc: LOGIc: FACT: (CH <x>|CH <x> _ D <x>), 2-1099
TRIGger (A|B): LOWerthresHold CH <x>, 2-1099
TRIGger (A|B): PULSEWidth HIGHLimit, 2-1099
TRIGger (A|B): PULSEWidth LOWlimit, 2-1099
TRIGGER: {A|B}: PULSEWidth: LOGICQUALification, 2-1096
TRIGGER: {A|B}: PULSEWidth LOWLimit, 2-1097
TRIGGER: {A|B}: PULSEWidth POLarity, 2-1097
TRIGGER: {A|B}: PULSEWidth SOUrce, 2-1098
TRIGGER: {A|B}: PULSEWidth WHEN, 2-1098
TRIGGER: {A|B}: RUNT: LOGICQUALification, 2-1099
TRIGGER: {A|B}: RUNT: POLarity, 2-1100
TRIGGER: {A|B}: RUNT: SOUrce, 2-1100
TRIGGER: {A|B}: RUNT: WHEN, 2-1101
TRIGGER: {A|B}: RUNT: WIDTH, 2-1102
TRIGGER: {A|B}: SETHOLD CLOCk EDGE, 2-1102
TRIGGER: {A|B}: SETHOLD CLOCk SOUrce, 2-1103
TRIGGER: {A|B}: SETHOLD HOLDTime, 2-1103
TRIGGER: {A|B}: SETHOLD SETTime, 2-1104
TRIGGER: {A|B}: SETHOLD LOGical: (CH<><|CH<><_D<><>), 2-1104
TRIGGER: {A|B}: TIMEOUT: LOGICQUALification, 2-1105
TRIGGER: {A|B}: TIMEOUT POLarity, 2-1106
TRIGGER: {A|B}: TIMEOUT SOUrce, 2-1106
TRIGGER: {A|B}: TIMEOUT TIME, 2-1107
TRIGGER: {A|B}: TRANsition DELTatime, 2-1107
TRIGGER: {A|B}: TRANsition LOGICAL: (CH<><|CH<><_D<><>), 2-1108
TRIGGER: {A|B}: TRANsition POLarity, 2-1108
TRIGGER: {A|B}: TRANsition SOUrce, 2-1109
TRIGGER: {A|B}: TRANsition WHEN, 2-1109
TRIGGER: {A|B}: TYPE, 2-1110
TRIGGER: {A|B}: UPPERthreshold CH<><, 2-1111
TRIGGER: {A|B}: WINDOW CROSSing, 2-1112
TRIGGER: {A|B}: WINDOW LOGICAL: (CH<><|CH<><_D<><>), 2-1113
TRIGGER: {A|B}: WINDOW SOUrce, 2-1113
TRIGGER: {A|B}: WINDOW WHEN, 2-1114
TRIGGER: {A|B}: WINDOW WIDTH, 2-1114
TRIGGER: {A|B}: B: RESET), 2-976
TRIGGER: A: HOLDoff BY, 2-1115
TRIGGER: A: HOLDoff TIME, 2-1115
TRIGGER: A: LOGICQUALification, 2-1116
TRIGGER: A: MODE, 2-1117
TRIGGER: AUXLevel, 2-1117
TRIGGER: B: BY, 2-1118
TRIGGER: B: EVENTS COUNT, 2-1118
TRIGGER: B: RESET, 2-1119
TRIGGER: B: RESET EDGE COUPling, 2-1119
TRIGGER: B: RESET EDGE LEVel, 2-1120
TRIGGER: B: RESET EDGE SLOPe, 2-1120
TRIGGER: B: RESET EDGE SOUrce, 2-1121
TRIGGER: B: RESET TIMEOut TIME, 2-1121
TRIGGER: B: RESET TYPE, 2-1122
TRIGGER: B: STATE, 2-1122
TRIGGER: B: TIME, 2-1123
TRIGGER: B: STATE?, 2-1123
*TST?, 2-1124

U
UNDO, 2-1124
UNLock, 2-1125
USB Device: CONFIGure, 2-1125

V
VERBose, 2-1126
VERTical DESKEw FROM: CUSTOMPROPAGation, 2-1127
VERTical DESKEw FROM SOUrce, 2-1127
VERTical DESKEw STATIC, 2-1128
VERTical DESKEw TO: CUSTOMPROPAGation, 2-1128
VERTical DESKEw TO SOUrce, 2-1129
VISual AREA<><: ASPECTratio, 2-1129
VISual AREA<><: FLIP HOrizontal, 2-1130
VISual AREA<><: FLIP VERTical, 2-1130
VISual AREA<><: HEIGHT, 2-1130
VISual AREA<><: HITType, 2-1131
VISual AREA<><: RESET, 2-1131
VISual AREA<><: ROTAtion, 2-1132
VISual AREA<><: SHAPE, 2-1132
VISual AREA<><: SOUrce, 2-1133
VISual AREA<><: VERTICES, 2-1133
VISual AREA<><: WIDTH, 2-1134
VISual AREA<><: XPOStion, 2-1135
VISual AREA<><: YPOStion, 2-1135
VISual DELETEALL, 2-1136
VISual ENAble, 2-1136
VISual EQUation, 2-1136
VISual SHOWArEas, 2-1137
VISual SHOWcriteria, 2-1137
VISual SHOWEquation, 2-1138

W
*WA1, 2-1139
Index

WAVFrm?, 2-1139
WFMOutpre?, 2-1140
WFMOutpre:ASC_Fmt?, 2-1140
WFMOutpre:BIT_Nr, 2-1141
WFMOutpre:BN_Fmt, 2-1141
WFMOutpre:BYT_Nr, 2-1142
WFMOutpre:BYT_Or, 2-1143
WFMOutpre:DOMain?, 2-1143
WFMOutpre:ENCdg, 2-1144
WFMOutpre:NR_Pt?, 2-1144
WFMOutpre:PT_FMT?, 2-1145
WFMOutpre:PT_Off?, 2-1145
WFMOutpre:WFId?, 2-1146
WFMOutpre:XINcr?, 2-1147
WFMOutpre:XUNit?, 2-1148
WFMOutpre:XZEr?, 2-1148
WFMOutpre:YMUlt?, 2-1149
WFMOutpre:YOFF?, 2-1149
WFMOutpre:YUNit?, 2-1150
WFMOutpre:YZEro?, 2-1150