MSO4000 and DPO4000 Series
Digital Phosphor Oscilloscopes
Programmer Manual
Copyright © Tektronix. All rights reserved. Licensed software products are owned by Tektronix or its subsidiaries or suppliers, and are protected by national copyright laws and international treaty provisions.

Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specifications and price change privileges reserved.

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

**Contacting Tektronix**

Tektronix, Inc.
14200 SW Karl Braun Drive
P.O. Box 500
Beaverton, OR 97077
USA

For product information, sales, service, and technical support:
- In North America, call 1-800-833-9200.
- Worldwide, visit www.tektronix.com to find contacts in your area.
Table of Contents

Getting Started .................................................................................................... 1-1
Setting Up Remote Communications ................................................................. 1-1
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
  Alias Command Group ................................................................................ 2-12
  Bus Command Group .................................................................................. 2-13
  Calibration and Diagnostic Command Group .............................................. 2-17
  Cursor Command Group .......................................................................... 2-17
  Display Command Group ............................................................................ 2-19
  Ethernet Command Group ........................................................................ 2-20
  File System Command Group ..................................................................... 2-21
  Hard Copy Command Group ....................................................................... 2-22
  Histogram Command Group ...................................................................... 2-24
  Horizontal Command Group ...................................................................... 2-24
  Mark Command Group ................................................................................. 2-25
  Math Command Group ............................................................................... 2-26
  Measurement Command Group .................................................................... 2-27
  Miscellaneous Command Group .................................................................. 2-30
  PictBridge Command Group ...................................................................... 2-32
  Power Command Group .............................................................................. 2-32
  Save and Recall Command Group ............................................................... 2-40
  Search Command Group ............................................................................. 2-41
  Status and Error Command Group ............................................................ 2-48
  Trigger Command Group ........................................................................... 2-49
  Vertical Command Group .......................................................................... 2-60
  Waveform Transfer Command Group .......................................................... 2-63
  Zoom Command Group .............................................................................. 2-68
Commands Listed in Alphabetical Order ............................................................ 2-71
Status and Events ............................................................................................. 3-1
  Registers ..................................................................................................... 3-1
  Queues ........................................................................................................ 3-4
  Event Handling Sequence ........................................................................... 3-5
  Synchronization Methods .......................................................................... 3-7
# Table of Contents

Appendix A: Character Set ........................................................................................................... A-1  
Appendix B: Reserved Words ....................................................................................................... B-1  
Appendix C: Factory Defaults 
  Default Setup ......................................................................................................................... C-1  
Appendix D: Programming Example ............................................................................................ D-1  
Index
List of Figures

Figure 3-1: The Standard Event Status Register (SESR) .................................................... 3-1
Figure 3-2: The Status Byte Register (SBR) ................................................................. 3-2
Figure 3-3: The Device Event Status Enable Register (DESER) .................................... 3-3
Figure 3-4: The Event Status Enable Register (ESER) .................................................. 3-3
Figure 3-5: The Service Request Enable Register (SRER) ......................................... 3-4
Figure 3-6: Status and Event Handling Process .............................................................. 3-6
Figure 3-7: Command Processing Without Using Synchronization ............................... 3-8
Figure 3-8: Processing Sequence With Synchronization .............................................. 3-8
List of Tables

Table 1-1: USB Device Parameters ........................................................................ 1-2
Table 2-1: Symbols for Backus-Naur Form .............................................................. 2-1
Table 2-2: Command Message Elements .................................................................. 2-2
Table 2-3: Comparison of Header Off and Header On Responses .............................. 2-3
Table 2-4: End of Message Terminator ..................................................................... 2-5
Table 2-5: Channel Mnemonics .............................................................................. 2-6
Table 2-6: Cursor Mnemonics ................................................................................. 2-6
Table 2-7: Math Specifier Mnemonics ...................................................................... 2-6
Table 2-8: Measurement Specifier Mnemonics ........................................................... 2-6
Table 2-9: Reference Waveform Mnemonics .............................................................. 2-6
Table 2-10: Numeric Arguments ............................................................................. 2-7
Table 2-11: Quoted String Argument ....................................................................... 2-7
Table 2-12: Block Argument .................................................................................... 2-8
Table 2-13: Acquisition Commands ........................................................................ 2-11
Table 2-14: Alias Commands ................................................................................. 2-12
Table 2-15: Bus Commands ................................................................................... 2-13
Table 2-16: Calibration and Diagnostic Commands .................................................. 2-17
Table 2-17: Cursor Commands ............................................................................... 2-18
Table 2-18: Display Commands ............................................................................. 2-20
Table 2-19: Ethernet Commands ............................................................................ 2-20
Table 2-20: File System Commands ....................................................................... 2-22
Table 2-21: Hard Copy Commands ......................................................................... 2-22
Table 2-22: Histogram Commands ......................................................................... 2-24
Table 2-23: Horizontal Commands ........................................................................ 2-24
Table 2-24: Mark Commands ................................................................................. 2-25
Table 2-25: Math Commands ............................................................................... 2-26
Table 2-26: Measurement Commands .................................................................... 2-27
Table 2-27: Miscellaneous Commands .................................................................. 2-30
Table 2-28: PictBridge Commands ....................................................................... 2-32
Table 2-29: Power Commands ............................................................................ 2-32
Table 2-30: Save and Recall Commands .................................................................. 2-40
Table 2-31: Status and Error Commands ................................................................. 2-48
Table 2-32: Trigger Commands ............................................................................ 2-49
Table 2-33: Vertical Commands ............................................................................. 2-60
Table 2-34: Binary data ranges .............................................................................. 2-64
Table 2-35: Waveform Transfer Commands ............................................................ 2-66
Table 2-36: Zoom Commands .............................................................................. 2-68
Table 2-37: Supported display formats ................................................................... 2-95
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-38</td>
<td>Channel Offset Range</td>
<td>2-126</td>
</tr>
<tr>
<td>2-39</td>
<td>DATa and WFMOutpre Parameter Settings</td>
<td>2-153</td>
</tr>
<tr>
<td>2-40</td>
<td>FPAnel:PRESS arguments</td>
<td>2-184</td>
</tr>
<tr>
<td>2-41</td>
<td>FPAnel:TURN arguments</td>
<td>2-185</td>
</tr>
<tr>
<td>2-42</td>
<td>Advanced Math expression elements</td>
<td>2-210</td>
</tr>
<tr>
<td>2-43</td>
<td>Available HDTV formats</td>
<td>2-490</td>
</tr>
<tr>
<td>2-44</td>
<td>Video Line Numbering Ranges</td>
<td>2-491</td>
</tr>
<tr>
<td>2-45</td>
<td>Waveform Suffixes</td>
<td>2-525</td>
</tr>
<tr>
<td>3-1</td>
<td>SESR Bit Functions</td>
<td>3-2</td>
</tr>
<tr>
<td>3-2</td>
<td>SBR Bit Functions</td>
<td>3-2</td>
</tr>
<tr>
<td>3-3</td>
<td>Oscilloscope operations that can generate OPC</td>
<td>3-7</td>
</tr>
<tr>
<td>3-4</td>
<td>No Event Messages</td>
<td>3-12</td>
</tr>
<tr>
<td>3-5</td>
<td>Command Error Messages (CME Bit 5)</td>
<td>3-13</td>
</tr>
<tr>
<td>3-6</td>
<td>Execution Error Messages (EXE Bit 4)</td>
<td>3-13</td>
</tr>
<tr>
<td>3-7</td>
<td>Device Error Messages (DDE Bit 3)</td>
<td>3-16</td>
</tr>
<tr>
<td>3-8</td>
<td>System Event Messages</td>
<td>3-16</td>
</tr>
<tr>
<td>3-9</td>
<td>Execution Warning Messages (EXE Bit 4)</td>
<td>3-17</td>
</tr>
<tr>
<td>3-10</td>
<td>Execution Warning Messages (EXE Bit 4)</td>
<td>3-17</td>
</tr>
<tr>
<td>3-11</td>
<td>Internal Warning Messages</td>
<td>3-17</td>
</tr>
<tr>
<td>C-1</td>
<td>Default values</td>
<td>C-1</td>
</tr>
</tbody>
</table>
Getting Started

This manual explains the use of commands for remotely controlling 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.

Setting Up Remote Communications

You can remotely communicate between your oscilloscope and PC via the Ethernet, USB, and, GPIB using the TEK-USB-488 Adapter.

**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 10/100 Base-T local area network.

To change the Ethernet settings on your oscilloscope, do the following:

1. On the front panel, push **Utility**.
2. Push **Utility Page**.
3. Select **I/O** with the Multipurpose knob.
4. Push **Ethernet Network Settings**.
5. On the side-bezel menu, if you are on a DHCP Ethernet network and using a through cable, set DHCP/BOOTP to **On**.
6. If you are using a cross-over cable, set DHCP/BOOTP to **Off**, and set a hard coded TCP/IP address.

**USB**

If you are using USB, start by connecting an appropriate USB cable to the USB 2.0 high-speed device port on the rear panel of your oscilloscope.
With USB, the system automatically configures itself. To verify that the USB is enabled:

1. On the front panel, push **Utility**.

2. Push **Utility Page**.

3. Select I/O with the Multipurpose knob.

4. Push **USB**, and verify that USB is enabled.

5. If USB is disabled, push **Connect to computer** on the side-bezel menu.

After connection, the host, with appropriate software, can list the oscilloscope as a USB device with the following parameters. (See Table 1-1.)

**Table 1-1: USB Device Parameters**

<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>0x0401 (decimal 1025)</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>

**GPIB**

To use GPIB, start by connecting an appropriate USB cable to the USB 2.0 high speed device port on the rear panel of your oscilloscope. Connect the other end to the TEK-USB-488 Adapter host port. Then connect a GPIB cable from the TEK-USB-488 Adapter to your PC.

Supply power to the Adapter in either of these two ways:

1. Use the optional 5 V$_{DC}$ power adapter connected to the 5 V$_{DC}$ power input on the Adapter.

2. Use an appropriate USB cable connected to a powered USB host port on your PC and the Device port on the TEK-USB-488 Adapter.
The oscilloscope has a USB 2.0 High speed device port to control the oscilloscope through USBTMC or GPIB with a TEK-USB-488 Adapter. The USBTMC protocol allows USB devices to communicate using IEEE488 style messages. This lets you run your GPIB software applications on USB hardware.

Before setting up the oscilloscope for remote communication using the electronic (physical) GPIB interface, you should familiarize yourself with the following GPIB requirements:

- A unique device address must be assigned to each device on the bus. No two devices can share the same device address.
- No than 15 devices can be connected to any one line.
- One device should be connected for every 6 feet (2 meters) of cable used.
- No than 65 feet (20 meters) of cable should be used to connect devices to a bus.
- At least two-thirds of the devices on the network should be powered on while using the network.
- Connect the devices on the network in a star or linear configuration. Do not use loop or parallel configurations.

To function correctly, your oscilloscope must have a unique device address. The default setting for the GPIB configuration is GPIB Address 1.

To change the GPIB address settings, do the following:

1. On the front panel, push Utility.
3. Select I/O with the Multipurpose knob.
4. Push GPIB.
5. Enter the GPIB address on the side-bezel menu, using the multipurpose knob. This will set the GPIB address on an attached TEK-USB-488 Adapter.

The oscilloscope is now set up for bidirectional communication with your controller.
The following documents are available for download on the Manuals Finder Web site at www.tektronix.com:

**MSO4000 and DPO4000 Series Digital Phosphor Oscilloscopes User Manual.** Information about installing and operating the oscilloscope.

**Getting Started with OpenChoice™ Solutions Manual.** Options for getting data from your oscilloscope into any one of several available analysis tools.

**MSO4000 and DPO4000 Series Digital Phosphor Oscilloscopes Technical Reference.** Oscilloscope specifications and a performance verification procedure.

**TekVISA Programmer Manual.** Description of TekVISA, the Tektronix implementation of the VISA Application Programming Interface (API). TekVISA is industry-compliant software for writing interoperable oscilloscope drivers in a variety of Application Development Environments (ADEs).
Command Syntax

You can control the operations and functions of the oscilloscope through the Ethernet port or the USB 2.0 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>
<tr>
<td>()</td>
<td>Comment</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 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>14:30:00</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:DOTsonly? will return the following.

   If the header is on:
   
   DISPLAY:GRATICULE FULL;:DISPLAY:STYLE:DOTSONLY 1
If the header is off:

```
FULL;1
```

5. 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:NORMal;ACQuire:NUMAVg 8 (no colon before ACQuire)
DISPlay:GRAticule FULL;:DOTSONLY OFF (extra colon before DOTSonly. You could use DISPlay:DOTSonly OFF instead)
DISPlay:GRAticule FULL;:*TRG (colon before a star (*) command)
MATH:HORizontal:SCAle 1.0e-1;HORizontal:POSition 5.0e1 (levels of the mnemonics are different; either remove the second use of HORIZontal: or place :MATH in front of HORIZontal: POSition)
```

**Terminating**

This documentation uses `<EOM>` (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><code>&lt;EOM&gt;</code></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, or CH4. 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<x>. 

MSO4000 and DPO4000 Series Programmer Manual 2-5
**Command Syntax**

- **Cursor Position Mnemonics**
  When cursors are displayed, commands may specify which cursor of the pair to use.

  **Table 2-5: 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 4.</td>
</tr>
</tbody>
</table>

  **Table 2-6: 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 either 1 or 2.</td>
</tr>
<tr>
<td>POSITION&lt;x&gt;</td>
<td>A cursor selector; &lt;x&gt; is either 1 or 2.</td>
</tr>
<tr>
<td>HPOS&lt;x&gt;</td>
<td>A cursor selector; &lt;x&gt; is either 1 or 2.</td>
</tr>
</tbody>
</table>

- **Math Specifier Mnemonics**
  Commands can specify the mathematical waveform to use as a mnemonic in the header.

  **Table 2-7: Math Specifier 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. Up to eight automated measurements may be displayed.

  **Table 2-8: 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 through 4.</td>
</tr>
</tbody>
</table>

- **Channel Mnemonics**
  Commands specify the channel to use as a mnemonic in the header.

- **Reference Waveform Mnemonics**
  Commands can specify the reference waveform to use as a mnemonic in the header.

  **Table 2-9: 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, 2, 3, or 4 for 4-channel oscilloscopes and 1 or 2 for 2-channel oscilloscopes.</td>
</tr>
</tbody>
</table>
Argument Types

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: `SAVe:WAVEform:FILEFormat INTERNaI`

Numeric

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 2-10: 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>Digital data 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 2-11: 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. If you use a GPIB network, you cannot terminate a quoted string with the END message before the closing delimiter.

6. 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.

7. 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

Several oscilloscope commands use a block argument form, as defined in the table below.

**Table 2-12: 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;...]&lt;DChar&gt;...][#0]&lt;DChar&gt;...&lt;terminator&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 Syntax

```
Alias:DEFINE "SETUp1",#231AUTOSet EXECute,:,:SELECT:REF1 ON
```

- **Block Header**: Specifies number of length digits that follow.
- **Block Argument**: Specifies data length.
Command Groups

This manual lists the MSO4000 and DPO4000 Series IEEE488.2 commands in two ways. First, it presents them by functional groups. Then, it lists them alphabetically. The functional group list starts below. The alphabetical list provides detail on each command. (See page 2-71, Commands Listed in Alphabetical Order.)

Acquisition Command Group

Use the commands in the Acquisition Command Group to set up the modes and functions that control how the oscilloscope acquires signals input to the channels, and processes them into waveforms.

Using the commands in this group, 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 channel waveforms.
■ Set acquisition parameters.

Table 2-13: Acquisition Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQuire?</td>
<td>Returns acquisition parameters</td>
</tr>
<tr>
<td>ACQuire:MAGnivu</td>
<td>Sets or returns the MagniVu feature</td>
</tr>
<tr>
<td>ACQuire:MAXSamplerate?</td>
<td>Returns the maximum real-time sample rate</td>
</tr>
<tr>
<td>ACQuire:MODE</td>
<td>Sets or returns the acquisition mode</td>
</tr>
<tr>
<td>ACQuire:NUMACq?</td>
<td>Returns number of acquisitions that have occurred</td>
</tr>
<tr>
<td>ACQuire:NUMAVg</td>
<td>Sets or returns the number of acquisitions for an averaged waveform</td>
</tr>
<tr>
<td>ACQuire:STATE</td>
<td>Starts or stops the acquisition system</td>
</tr>
<tr>
<td>ACQuire:STOPAfer</td>
<td>Sets or returns whether the acquisition is continuous or single sequence</td>
</tr>
</tbody>
</table>
Alias Command Group

Use the Alias commands to define new commands as a sequence of standard commands. You may 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 IEEE488.2 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 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.

Table 2-14: Alias Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALIas</td>
<td>Sets or returns 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 returns the alias state</td>
</tr>
</tbody>
</table>
Bus Command Group

Use the Bus commands when working with serial or parallel bus measurements.

- Install the DPOEMBD application module when working with I²C or SPI bus signals.
- Install the DPO4COMP module when working with RS-232, RS-422, RS-485, or UART bus signals.
- Install the DPO4AUDIO module when working with I²S, Left Justified (LJ), Right Justified (RJ), or TDM bus signals.
- Install the DPO4AUTO or DPO4AUTOMAX module when working with CAN or LIN bus signals.
- Install the DPO4AUTOMAX module when working with FlexRay bus signals.

**NOTE.** LIN and FlexRay work on DPO4000s with serial numbers greater than C020000 and on all MSO4000s.

**NOTE.** The parallel bus commands work with MSO4000 Series oscilloscopes only.

**NOTE.** The Search Command Group and the Trigger Command Group also contain bus-related commands.

**Table 2-15: Bus Commands**

<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS?</td>
<td>Returns the parameters for each bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:BITDelay</td>
<td>Sets or returns the number of delay bits for the AUDIO bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:BITOrder</td>
<td>Sets or returns the bit order for the AUDIO bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:CHANnel:SIZe</td>
<td>Sets or returns the number of bits per channel for the AUDIO bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:CLOCk:POLarity</td>
<td>Sets or returns the clock polarity for the AUDIO bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:CLOCk:SOUrce</td>
<td>Sets or returns the clock source waveform for the AUDIO bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:DATa:POLarity</td>
<td>Sets or returns the data polarity for the AUDIO bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:DATa:SIZe</td>
<td>Sets or returns the number of bits per word for the AUDIO bus</td>
</tr>
<tr>
<td>Commands</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:DATa:SOUrce</td>
<td>Sets or returns the data source waveform for the AUDIO bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:DISplay:FORMat</td>
<td>Sets or returns the display format for the AUDIO bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:FRAME:SIZe</td>
<td>Sets or returns the number of channels in each frame for the AUDIO bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:FRAMESync:POLarity</td>
<td>Sets or returns the frame sync polarity for the AUDIO bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:FRAMESync:SOUrce</td>
<td>Sets or returns the frame sync source waveform for the AUDIO bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:TYPE</td>
<td>Sets or returns the audio format (type) for the AUDIO bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:WORDSSel:POLarity</td>
<td>Sets or returns the word select polarity for the AUDIO bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:AUDio:WORDSSel:SOUrce</td>
<td>Sets or returns the word select source waveform for the AUDIO bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:CAN:BITRate</td>
<td>Sets or returns the bit rate for the CAN bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:CAN:PRObe</td>
<td>Sets or returns the probing method used to probe the CAN bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:CAN:SAMPLEpoint</td>
<td>Sets or returns the sample point (in %) to sample during each bit period</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:CAN:SOUrce</td>
<td>Sets or returns the CAN data source</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:DISplay:FORMat</td>
<td>Sets the display format for the numerical information in the specified bus waveform</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:DISplay:TYPE</td>
<td>Sets the display type for the specified bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:FLEXray:BITRate</td>
<td>Sets or returns the bit rate for FLEXRAY</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:FLEXray:Channel</td>
<td>Sets or returns the FLEXRAY ID format</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:FLEXray:SIGnal</td>
<td>Sets or returns the FLEXRAY standard</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:FLEXray:SOUrce</td>
<td>Sets or returns the FLEXRAY data source</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:I2C:ADDRes:RWINClude</td>
<td>Sets and returns whether the read/write bit is included in the address</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:I2C:(CLOCK</td>
<td>SCLK):SOUrce</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:I2C:(DATA</td>
<td>SDATA):SOUrce</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:LABel</td>
<td>Sets or returns the waveform label for the specified bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:LIN:BITRate</td>
<td>Sets or returns the bit rate for LIN</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:LIN:IDFORmat</td>
<td>Sets or returns the LIN ID format</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:LIN:POLARity</td>
<td>Sets or returns the LIN polarity</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:LIN:SAMPLEpoint</td>
<td>Sets or returns the sample point (in %) at which to sample during each bit period</td>
</tr>
<tr>
<td>Commands</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:LIN:SOUrce</td>
<td>Sets or returns the LIN data source</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:LIN:STANDard</td>
<td>Sets or returns the LIN standard</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:PARallel:BIT&lt;x&gt;:SOUrce</td>
<td>Sets or returns the Parallel bit source for the specified bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:PARallel:CLOCK:EDGE</td>
<td>Sets or returns the Parallel clock edge for the specified bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:PARallel:CLOCK:ISCLKed</td>
<td>Sets or returns the Parallel bus clock function for the specified bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:PARallel:CLOCK:SOUrce</td>
<td>Sets or returns the Parallel clock source for the specified bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:PARallel:WIDTH</td>
<td>Sets or returns the width of the Parallel bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:POSition</td>
<td>Sets or returns the position of the specified bus waveform</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:BITRate</td>
<td>Sets or returns the RS-232 bit rate for the specified bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:DATABits</td>
<td>Sets or returns the number of bits for the data frame</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:DELIMiter</td>
<td>Sets or returns 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 returns the display mode for the specified bus display and event table</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:PARity</td>
<td>Sets or returns parity for RS-232 data</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:POLarity</td>
<td>Sets or returns the RS-232C polarity for the specified bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:RX:SOUrce</td>
<td>Sets or returns the RS-232 RX source</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:RS232C:TX:SOUrce</td>
<td>Sets or returns the RS-232 TX Source</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:BITOrder</td>
<td>Sets or returns the bit order for the specified SPI bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:{CLOCK</td>
<td>:SCLK}:POLARity</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:{CLOCK</td>
<td>:SCLK}:SOUrce</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:DATA:{IN</td>
<td>:MISO}:POLARity</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:DATA:{IN</td>
<td>:MISO}:SOUrce</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:DATA:{OUT</td>
<td>:MOSI}:POLARity</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:DATA:{OUT</td>
<td>:MOSI}:SOUrce</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:DATA:SIZe</td>
<td>Sets or returns the number of bits per word for the specified bus</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:FRAMING</td>
<td>Sets or returns the SPI framing type</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI:IDLETime</td>
<td>Sets or returns the SPI bus idle time in seconds for the specified bus</td>
</tr>
</tbody>
</table>
### Table 2-15: Bus Commands (cont.)

<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS:B&lt;x&gt;:SPI{:SELect</td>
<td>:SS}:POLARity</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:SPI{:SELect</td>
<td>:SS}:SOUrce</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:STATE</td>
<td>Turns the specified bus on and off</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:TYPE</td>
<td>Sets or returns the specified bus type</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:USB:BITRate</td>
<td>Sets or returns the USB bit rate</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:USB:PRObe</td>
<td>Sets or returns the type of probes attached to the USB signal</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:USB:SOUrce:DIFFerential</td>
<td>Sets or returns the USB source channels when using a differential probe</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:USB:SOUrce:DMINus</td>
<td>Sets or returns the USB D- source channel</td>
</tr>
<tr>
<td>BUS:B&lt;x&gt;:USB:SOUrce:DPLUS</td>
<td>Sets or returns the USB D+ source channel</td>
</tr>
<tr>
<td>BUS:LOWerthreshold:CH&lt;x&gt;</td>
<td>Sets or returns the lower threshold for each channel</td>
</tr>
<tr>
<td>BUS:THReshold:CH&lt;x&gt;</td>
<td>Sets or returns the threshold for a channel</td>
</tr>
<tr>
<td>BUS:THReshold:D&lt;x&gt;</td>
<td>Sets or returns the threshold for a digital channel</td>
</tr>
<tr>
<td>BUS:UPPerthreshold:CH&lt;x&gt;</td>
<td>Sets or returns the upper threshold for each channel</td>
</tr>
</tbody>
</table>
Calibration and Diagnostic Command Group

The Calibration and Diagnostic commands provide information about the current state of oscilloscope calibration. They also initiate internal signal path calibration (SPC) or execute diagnostic tests. Commands that are specific to factory calibration are not described in this manual. They are described in the Service manual, located on the DPO4000 Documentation CD-ROM in PDF format. You can also order a printed copy.

Table 2-16: Calibration and Diagnostic Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*CAL?</td>
<td>Instructs the oscilloscope to perform self-calibration and returns the oscilloscope self calibration status</td>
</tr>
<tr>
<td>CALibrate:FACTory:STATus?</td>
<td>Returns the factory calibration status value saved in nonvolatile memory</td>
</tr>
<tr>
<td>CALibrate:INTERNal</td>
<td>Starts a signal path compensation</td>
</tr>
<tr>
<td>CALibrate:INTERNal:STARt</td>
<td>Starts the internal signal path calibration</td>
</tr>
<tr>
<td>CALibrate:INTERNal:STATus?</td>
<td>Returns the current status of the internal signal path calibration</td>
</tr>
<tr>
<td>CALibrate:RESULTS?</td>
<td>Returns the status of all calibration subsystems without performing an SPC operation</td>
</tr>
<tr>
<td>CALibrate:RESULTS:FACTory?</td>
<td>Returns the status of internal and factory calibration</td>
</tr>
<tr>
<td>CALibrate:RESULTS:SPC?</td>
<td>Returns the results of the last SPC operation</td>
</tr>
<tr>
<td>DIAg:LOOP:OPTion</td>
<td>Sets the self-test loop option</td>
</tr>
<tr>
<td>DIAg:LOOP:OPTion:NTIMes</td>
<td>Sets the self-test loop option to run N times</td>
</tr>
<tr>
<td>DIAg:LOOP:STOP</td>
<td>Stops the self-test at the end of the current loop</td>
</tr>
<tr>
<td>DIAg:RESULT:FLAg?</td>
<td>Returns the pass/fail status from the last self-test sequence execution</td>
</tr>
<tr>
<td>DIAg:RESULT:LOG?</td>
<td>Returns the internal results log from the last self-test sequence execution</td>
</tr>
<tr>
<td>DIAg:SELECT:&lt;function&gt;</td>
<td>Selects one of the available self-test areas</td>
</tr>
<tr>
<td>DIAg:STATE</td>
<td>Sets the oscilloscope operating state</td>
</tr>
<tr>
<td>DIAg:SELECT</td>
<td>Runs self tests on the specified system subsystem</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 cursor 1 and cursor 2, such as cursor position.
You can also use the commands to select one of the following cursor functions:

- **Off.** Turns off the display of all cursors.
- **Waveform Cursors.** Consists of two cursors. Waveform cursors enable you to conveniently measure waveform amplitude and time.
- **Screen Cursors.** Consists of two pairs of independent horizontal and vertical cursors. You can use these cursors to indicate an arbitrary position within the waveform display area.

### Table 2-17: Cursor Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURSor?</td>
<td>Returns cursor settings</td>
</tr>
<tr>
<td>CURSor:DDT?</td>
<td>Returns the cursor deltaY/deltaT (dY/dT) readout</td>
</tr>
<tr>
<td>CURSor:FUNCTION</td>
<td>Sets or returns the cursor type</td>
</tr>
<tr>
<td>CURSor:HBars?</td>
<td>Returns hbar cursor settings</td>
</tr>
<tr>
<td>CURSor:HBars:DELTa?</td>
<td>Returns hbars cursors vertical difference</td>
</tr>
<tr>
<td>CURSor:HBars:POSITION&lt;x&gt;</td>
<td>Sets or returns the hbar cursor&lt;x&gt; vertical position</td>
</tr>
<tr>
<td>CURSor:HBars:UNIts</td>
<td>Returns hbar cursor units</td>
</tr>
<tr>
<td>CURSor:HBars:USE</td>
<td>Sets the horizontal bar cursor measurement scale, for use with ratio cursors</td>
</tr>
<tr>
<td>CURSor:MODE</td>
<td>Sets or returns whether cursors move in unison or separately</td>
</tr>
<tr>
<td>CURSor:VBars?</td>
<td>Sets or returns the position of vertical bar cursors</td>
</tr>
<tr>
<td>CURSor:VBars:ALTERNATE&lt;x&gt;</td>
<td>Returns the alternate readout for the waveform (Vbar) cursors</td>
</tr>
<tr>
<td>CURSor:VBars:DELTa?</td>
<td>Returns the horizontal difference between vbar cursors</td>
</tr>
<tr>
<td>CURSor:VBars:HPOS&lt;x&gt;?</td>
<td>Returns the vertical value of the specified vertical bar tick</td>
</tr>
<tr>
<td>CURSor:VBars:POSITION&lt;x&gt;</td>
<td>Sets or returns the vbar cursor&lt;x&gt; horizontal position</td>
</tr>
<tr>
<td>CURSor:VBars:UNIts</td>
<td>Sets or returns the horizontal units for vbar cursors</td>
</tr>
<tr>
<td>CURSor:VBars:USE</td>
<td>Sets the vertical bar cursor measurement scale</td>
</tr>
<tr>
<td>CURSor:VBars:VDELTa?</td>
<td>Returns the vertical difference between the two vertical bar cursor ticks</td>
</tr>
<tr>
<td>CURSor:XY:POLar:RADIUS:DELta?</td>
<td>Returns the difference between the cursors X radius and the cursor Y radius</td>
</tr>
</tbody>
</table>
### Table 2-17: Cursor Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURSor:XY:POLar:RADIUS:POSITION&lt;x&gt;?</td>
<td>Returns the polar radius of the specified cursor</td>
</tr>
<tr>
<td>CURSor:XY:POLar:RADIUS:UNIts?</td>
<td>Returns the polar radius units</td>
</tr>
<tr>
<td>CURSor:XY:POLar:THETA:DELta?</td>
<td>Returns the XY cursor polar coordinate delta</td>
</tr>
<tr>
<td>CURSor:XY:POLar:THETA:POSITION&lt;x&gt;?</td>
<td>Returns the cursor X or cursor Y polar coordinate</td>
</tr>
<tr>
<td>CURSor:XY:POLar:THETA:UNIts?</td>
<td>Returns the cursor polar coordinate units</td>
</tr>
<tr>
<td>CURSor:XY:PRODUCT:DELta?</td>
<td>Returns the difference between the cursors X position and cursor Y position</td>
</tr>
<tr>
<td>CURSor:XY:PRODUCT:POSITION&lt;x&gt;?</td>
<td>Returns the position of the X or Y cursor used to calculate the X × Y cursor measurement</td>
</tr>
<tr>
<td>CURSor:XY:PRODUCT:UNIts?</td>
<td>Returns the XY cursor product units</td>
</tr>
<tr>
<td>CURSor:XY:RATIO:DELta?</td>
<td>Returns the ratio of the difference between the cursor X position and cursor Y position</td>
</tr>
<tr>
<td>CURSor:XY:RATIO:POSITION&lt;x&gt;?</td>
<td>Returns the X or Y position for the specified cursor</td>
</tr>
<tr>
<td>CURSor:XY:RATIO:UNIts?</td>
<td>Returns the X and Y cursor units for the ratio measurement</td>
</tr>
<tr>
<td>CURSor:XY:READOUT</td>
<td>Sets or returns the XY cursor readout selection.</td>
</tr>
<tr>
<td>CURSor:XY:RECTangular:X:DELta?</td>
<td>Returns the cursor X delta value in rectangular coordinates</td>
</tr>
<tr>
<td>CURSor:XY:RECTangular:X:POSITION&lt;x&gt;</td>
<td>Sets or returns the cursor X rectangular coordinates</td>
</tr>
<tr>
<td>CURSor:XY:RECTangular:X:UNIts?</td>
<td>Returns the Cursor X rectangular units</td>
</tr>
<tr>
<td>CURSor:XY:RECTangular:Y:DELta?</td>
<td>Returns The cursor Y delta value in rectangular coordinates</td>
</tr>
<tr>
<td>CURSor:XY:RECTangular:Y:POSITION&lt;x&gt;&gt;</td>
<td>Sets or returns the cursor Y rectangular coordinate</td>
</tr>
<tr>
<td>CURSor:XY:RECTangular:Y:UNIts?</td>
<td>Returns the cursor Y rectangular units</td>
</tr>
</tbody>
</table>

### Display Command Group

Use the commands in the Display Command Group to change the graticule style, the display intensities, and to set the characteristics of the waveform display.

**NOTE.** *Your settings globally affect all displayed waveforms.*
Table 2-18: Display 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:CLOCk</td>
<td>Sets or returns the display of the date/time stamp</td>
</tr>
<tr>
<td>DISplay:DIGital:HEIght</td>
<td>Sets or returns the number of available digital waveform position slots</td>
</tr>
<tr>
<td>DISplay:GRAticule</td>
<td>Sets or returns the type of graticule that is displayed</td>
</tr>
<tr>
<td>DISplay:FORMat</td>
<td>Sets or returns the display format</td>
</tr>
<tr>
<td>DISplay:INTENSITy?</td>
<td>Returns all display intensity settings</td>
</tr>
<tr>
<td>DISplay:INTENSITy:BACKLight</td>
<td>Sets or returns the backlight intensity for the display</td>
</tr>
<tr>
<td>DISplay:INTENSITy:GRAticule</td>
<td>Sets or returns the graticule intensity for the display</td>
</tr>
<tr>
<td>DISplay:INTENSITy:WAVEform</td>
<td>Sets or returns the intensity of the waveforms</td>
</tr>
<tr>
<td>DISplay:PERSistence</td>
<td>Sets or returns display persistence setting</td>
</tr>
<tr>
<td>DISplay:STYLE:DOTsonly</td>
<td>Sets a dots-only display</td>
</tr>
<tr>
<td>MESSage</td>
<td>Sets or queries message box (screen annotation) parameters</td>
</tr>
<tr>
<td>MESSage:BOX</td>
<td>Sets or returns the coordinates of the message box</td>
</tr>
<tr>
<td>MESSage:CLEAR</td>
<td>Clears the contents of the message box</td>
</tr>
<tr>
<td>MESSage:SHOW</td>
<td>Sets or returns the contents of the message box</td>
</tr>
<tr>
<td>MESSage:STATE</td>
<td>Controls the display of the message box</td>
</tr>
</tbody>
</table>

**Ethernet Command Group**

Use the commands in the Ethernet Command Group to set up the Ethernet remote interface.

Table 2-19: Ethernet Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETHERnet:DHCPrebootp</td>
<td>Sets or returns the network initialization search for a DHCP/BOOTP server</td>
</tr>
<tr>
<td>ETHERnet:DNS:IPADDress</td>
<td>Sets or returns the network Domain Name Server (Dns) IP address</td>
</tr>
<tr>
<td>ETHERnet:DOMAINname</td>
<td>Sets or returns the network domain name</td>
</tr>
<tr>
<td>ETHERnet:ENETADDress?</td>
<td>Returns the Ethernet address value assigned to the oscilloscope</td>
</tr>
</tbody>
</table>
Table 2-19: Ethernet Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETHERnet:GATEWay:IPADDress</td>
<td>Sets or returns the remote interface gateway IP address</td>
</tr>
<tr>
<td>ETHERnet:HTTPPort</td>
<td>Sets or returns the remote interface HTTP port value</td>
</tr>
<tr>
<td>ETHERnet:IPADDress</td>
<td>Sets or returns the IP address assigned to the oscilloscope</td>
</tr>
<tr>
<td>ETHERnet:NAME</td>
<td>Sets or returns the network name assigned to the oscilloscope</td>
</tr>
<tr>
<td>ETHERnet:PASSWord</td>
<td>Sets or returns the Ethernet access password</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 pinging the gateway IP address</td>
</tr>
<tr>
<td>ETHERnet:SUBNETMask</td>
<td>Sets or returns the remote interface subnet mask value</td>
</tr>
</tbody>
</table>

File System Command Group

Use the commands in the File System Command Group to help you use CompactFlash and USB media. You can use the commands to do the following:

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

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

- File arguments are always enclosed within double quotes: "D:/MYDIR/TEK00001.SET"
- File names follow the non-case sensitive, MSDOS format: [DRIVE:]\[\PATH\]filename
- Path separators may be either forward slashes (/) or back slashes (\)
NOTE. Using back slash as a path separator may produce some unexpected results, depending on how your application treats escaped characters. Many applications recognize the sequence of 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" may be interpreted as a newline character; "\t" may be interpreted as a tab character. To ensure that this interpretation does not occur, you can use double backslashes. For example, "D:\testfile.txt".

Table 2-20: File System Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILESystem?</td>
<td>Returns the file system state</td>
</tr>
<tr>
<td>FILESystem:CWD</td>
<td>Sets or returns the current working directory for FILESystem commands</td>
</tr>
<tr>
<td>FILESystem:DELETE</td>
<td>Deletes a named file or directory</td>
</tr>
<tr>
<td>FILESystem:DIR?</td>
<td>Returns a list of directory contents</td>
</tr>
<tr>
<td>FILESystem:FORMat</td>
<td>Formats a named drive</td>
</tr>
<tr>
<td>FILESystem:FREESpace?</td>
<td>Returns the number of bytes of free space on the current drive</td>
</tr>
<tr>
<td>FILESystem:LDIR?</td>
<td>Returns a semicolon separated list of every file and directory in a folder.</td>
</tr>
<tr>
<td>FILESystem:MKDir</td>
<td>Creates a new directory</td>
</tr>
<tr>
<td>FILESystem:READFile</td>
<td>Writes the contents of the specified file to the specified interface</td>
</tr>
<tr>
<td>FILESystem:REName</td>
<td>Assigns a new name to an existing file</td>
</tr>
<tr>
<td>FILESystem:RMDir</td>
<td>Deletes a named directory</td>
</tr>
<tr>
<td>FILESystem:WRITEFile</td>
<td>Writes the specified block data to the oscilloscope current working directory</td>
</tr>
</tbody>
</table>

Hard Copy Command Group

Use the commands in the Hard Copy Command Group to make hard copies. PictBridge commands belong to a separate group. (See page 2-32, *PictBridge Command Group*.)

Table 2-21: Hard Copy Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HARDCopy</td>
<td>Sends a copy of the screen display to the selected printer</td>
</tr>
<tr>
<td>HARDCopy:ACTIVeprinter</td>
<td>Sets or returns the currently active printer</td>
</tr>
</tbody>
</table>
### Table 2-21: Hard Copy Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HARDCopy:INKSaver</td>
<td>Changes hard copy output to print color traces and graticule on a white background</td>
</tr>
<tr>
<td>HARDCopy:LAYout</td>
<td>Sets or returns the page orientation for hard copy</td>
</tr>
<tr>
<td>HARDCopy:PREVIEW</td>
<td>Previews the current screen contents with the InkSaver palette applied</td>
</tr>
<tr>
<td>HARDCopy:PRINTer:ADD</td>
<td>Adds a network printer to the list of available printers</td>
</tr>
<tr>
<td>HARDCopy:PRINTer:DELETE</td>
<td>Removes a network printer from the list of available printers</td>
</tr>
<tr>
<td>HARDCopy:PRINTer:LIST?</td>
<td>Returns the list of currently attached printers</td>
</tr>
<tr>
<td>HARDCopy:PRINTer:REName</td>
<td>Renames a network printer in the list of available printers</td>
</tr>
</tbody>
</table>
Histogram Command Group

Use the commands in the Histogram Command Group to set up a histogram for measurements.

Table 2-22: Histogram Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIStogram?</td>
<td>Returns all histogram parameters</td>
</tr>
<tr>
<td>HIStogram:BOX</td>
<td>Sets or returns the histogram box coordinates</td>
</tr>
<tr>
<td>HIStogram:COUNT</td>
<td>Clears the histogram count and statistics</td>
</tr>
<tr>
<td>HIStogram:DATA?</td>
<td>Returns the histogram data</td>
</tr>
<tr>
<td>HIStogram:DISPLAY</td>
<td>Sets or returns the scaling of the histogram display</td>
</tr>
<tr>
<td>HIStogram:END?</td>
<td>Returns the time of the last bin of the histogram</td>
</tr>
<tr>
<td>HIStogram:MODE</td>
<td>Sets or returns the type of the histogram, vertical or horizontal</td>
</tr>
<tr>
<td>HIStogram:SOURce</td>
<td>Sets or queries the source used to create the histogram</td>
</tr>
<tr>
<td>HIStogram:START?</td>
<td>Returns the time of the first bin of the histogram</td>
</tr>
</tbody>
</table>

Horizontal Command Group

Use the commands in the Horizontal Command Group to control the oscilloscope horizontal parameters.

Table 2-23: Horizontal Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HORizontal?</td>
<td>Returns settings for the horizontal commands</td>
</tr>
<tr>
<td>HORizontal:DELay:MODE</td>
<td>Sets or returns the horizontal delay mode</td>
</tr>
<tr>
<td>HORizontal:DELay:TIME</td>
<td>Sets or returns the horizontal delay time</td>
</tr>
<tr>
<td>HORizontal:DIGital:RECOrdlength:MAGnivu?</td>
<td>Returns the record length of the Magnivu digital acquisition</td>
</tr>
<tr>
<td>HORizontal:DIGital:RECOrdlength:MAIN?</td>
<td>Returns the record length of the main digital acquisition</td>
</tr>
<tr>
<td>HORizontal:DIGital:SAMPLERate:MAGnivu?</td>
<td>Returns the sample rate of the Magnivu digital acquisition</td>
</tr>
<tr>
<td>HORizontal:DIGital:SAMPLERate:MAIN?</td>
<td>Returns the sample rate of the main digital acquisition</td>
</tr>
<tr>
<td>HORizontal:POSITION</td>
<td>Sets or returns the horizontal position, in percent, that is used when delay is off</td>
</tr>
</tbody>
</table>
Table 2-23: Horizontal Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HORizontal:PREViewstate?</td>
<td>Returns the display system preview state</td>
</tr>
<tr>
<td>HORizontal:RECORdlength</td>
<td>Sets or returns the record length</td>
</tr>
<tr>
<td>HORizontal:SAMPLERate</td>
<td>Sets or returns the sample rate</td>
</tr>
<tr>
<td>HORizontal:SCAle</td>
<td>Sets or returns the horizontal scale</td>
</tr>
</tbody>
</table>

**Mark Command Group**

Use the commands in the Mark Command Group to identify areas of the acquired waveform that warrant further investigation.

Table 2-24: Mark Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MARK</td>
<td>Move to the next or previous mark on the waveform or returns all learnable settings from the mark commands</td>
</tr>
<tr>
<td>MARK:CREATE</td>
<td>Creates a mark on a particular waveform or all waveforms in a column</td>
</tr>
<tr>
<td>MARK:DELETE</td>
<td>Deletes a mark on a particular waveform, all waveforms in a column, or all marks</td>
</tr>
<tr>
<td>MARK:FREE?</td>
<td>Returns how many marks are free to be used</td>
</tr>
<tr>
<td>MARK:SELECTed:END?</td>
<td>Returns the end of the selected mark, in terms of 0 to 100% of the waveform</td>
</tr>
<tr>
<td>MARK:SELECTed:FOCUS?</td>
<td>Returns the focus of the selected mark, in terms of 0 to 100% of the waveform</td>
</tr>
<tr>
<td>MARK:SELECTed:MARKSINCOLUMN?</td>
<td>Returns how many marks are in the current zoom pixel column</td>
</tr>
<tr>
<td>MARK:SELECTed:OWNER?</td>
<td>Returns the owner of the selected mark</td>
</tr>
<tr>
<td>MARK:SELECTed:SOURCE?</td>
<td>Returns the source waveform of the selected mark</td>
</tr>
<tr>
<td>MARK:SELECTed:START?</td>
<td>Returns the start of the selected mark, in terms of 0 to 100% of the waveform</td>
</tr>
<tr>
<td>MARK:SELECTed:STATE?</td>
<td>Returns the on or off state of the selected mark</td>
</tr>
<tr>
<td>MARK:SELECTed:ZOOM:POSITION?</td>
<td>Returns the position of the selected mark, in terms of 0 to 100% of the upper window</td>
</tr>
<tr>
<td>MARK:TOTAL?</td>
<td>Returns how many marks are used</td>
</tr>
</tbody>
</table>
Math Command Group

Use the commands in the Math Command Group to create and define a math waveform. 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 waveform 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 up to 128 characters and comprising many sources, functions, and operands.

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH[1]?</td>
<td>Returns the definition of the math waveform</td>
</tr>
<tr>
<td>MATH[1]:DEFINE</td>
<td>Sets or returns the current math function as a text string</td>
</tr>
<tr>
<td>MATH[1]:HORIZONTAL:SCALE</td>
<td>Sets or returns the math horizontal display scale for FFT or for Dual Math waveforms</td>
</tr>
<tr>
<td>MATH[1]:HORIZONTAL:UNITS</td>
<td>Returns the math waveform horizontal unit value</td>
</tr>
<tr>
<td>{MATH</td>
<td>MATH1}:LABEL</td>
</tr>
<tr>
<td>MATH[1]:SPECTRAL:MAG</td>
<td>Sets or returns the units of spectral magnification in the math string</td>
</tr>
<tr>
<td>MATH[1]:SPECTRAL:WINDOW</td>
<td>Sets or returns the window function for math waveform spectral input data</td>
</tr>
<tr>
<td>MATH[1]:VERTICAL:POSITION</td>
<td>Sets or returns the vertical position of the currently selected math type</td>
</tr>
<tr>
<td>MATH[1]:VERTICAL:SCALE</td>
<td>Sets or returns the vertical scale of the currently selected math type</td>
</tr>
<tr>
<td>MATH[1]:VERTICAL:UNITS</td>
<td>Returns the math waveform vertical units</td>
</tr>
<tr>
<td>MATH[1]:HORIZONTAL:POSITION</td>
<td>Sets or returns the math horizontal display position for FFT or (non-live) math reference waveforms</td>
</tr>
<tr>
<td>MATH[1]:TYPE</td>
<td>Sets or returns the math waveform mode type</td>
</tr>
</tbody>
</table>
### Measurement Command Group

Use the commands in the Measurement Command Group to control the automated measurement system.

Up to four automated measurements can be displayed on the screen. In the commands, these measurement readouts are named `MEAS<x>`, where `<x>` is the measurement number.

In addition to the four displayed measurements, the measurement commands let you specify an additional measurement, IMMed. The immediate measurement has no front-panel equivalent. Immediate measurements are never displayed. Because they are computed only when needed, immediate measurements slow the waveform update rate less than displayed measurements.

Whether you use displayed or immediate measurements, use the `VALue` query to obtain measurement results.

Measurement commands can set and query measurement parameters. You can assign some parameters, such as waveform sources, differently for each measurement. Other parameters, such as reference levels, have only one value, which applies to all measurements.

#### Table 2-26: Measurement Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MEASUrement?</code></td>
<td>Returns all measurement parameters</td>
</tr>
<tr>
<td><code>MEASUrement:CLEARSNapshot</code></td>
<td>Removes the measurement snapshot display</td>
</tr>
<tr>
<td><code>MEASUrement:GATing</code></td>
<td>Sets or returns the measurement gating</td>
</tr>
<tr>
<td><code>MEASUrement:IMMed?</code></td>
<td>Returns all immediate measurement setup parameters</td>
</tr>
<tr>
<td><code>MEASUrement:IMMed:DELay?</code></td>
<td>Returns information about the immediate delay measurement</td>
</tr>
<tr>
<td><code>MEASUrement:IMMed:DELay:DIREction</code></td>
<td>Sets or returns the search direction to use for immediate delay measurements</td>
</tr>
<tr>
<td><code>MEASUrement:IMMed:DELay:EDGE&lt;x&gt;</code></td>
<td>Sets or returns the slope of the edge used for immediate delay “from” and “to” waveform measurements</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MEASUrement:IMMed:SOURce&lt;x&gt;</td>
<td>Sets or returns the “from” source for all single channel immediate measurements</td>
</tr>
<tr>
<td></td>
<td>Sets or returns the source to measure “to” for two-channel measurements</td>
</tr>
<tr>
<td>MEASUrement:IMMed:TYPE</td>
<td>Sets or returns the type of the immediate measurement</td>
</tr>
<tr>
<td>MEASUrement:IMMed:UNItst?</td>
<td>Returns the units of the immediate measurement</td>
</tr>
<tr>
<td>MEASUrement:IMMed:VALue?</td>
<td>Returns the value of the immediate measurement</td>
</tr>
<tr>
<td>MEASUrement:INDICators?</td>
<td>Returns all measurement indicator parameters</td>
</tr>
<tr>
<td>MEASUrement:INDICators:HORZ&lt;x&gt;??</td>
<td>Returns the position of the specified horizontal measurement indicator</td>
</tr>
<tr>
<td>MEASUrement:INDICators:NUMHORZ?</td>
<td>Returns the number of horizontal measurement indicators currently being displayed</td>
</tr>
<tr>
<td>MEASUrement:INDICators:NUMVERT?</td>
<td>Returns the number of vertical measurement indicators currently being displayed</td>
</tr>
<tr>
<td>MEASUrement:INDICators:STATE</td>
<td>Sets or returns the state of visible measurement indicators</td>
</tr>
<tr>
<td>MEASUrement:INDICators:VERT&lt;x&gt;??</td>
<td>Returns the value of the specified vertical measurement indicator</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;??</td>
<td>Returns all measurement parameters</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:COUNt?</td>
<td>Returns the number of values accumulated since the last statistical reset</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:DELaY?</td>
<td>Returns the delay measurement parameters for the specified measurement</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:DELaY:DIRection</td>
<td>Sets or returns the search direction to use for delay measurements</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:DELaY:EDGE&lt;x&gt;</td>
<td>Sets or returns the slope of the edge to use for delay “from” and “to” waveform measurements</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:MAXimum?</td>
<td>Returns the maximum value found since the last statistical reset</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:MEAN?</td>
<td>Returns the mean value accumulated since the last statistical reset</td>
</tr>
<tr>
<td>MEASUrement:MEAS&lt;x&gt;:MiNimum?</td>
<td>Returns the minimum value found since the last statistical reset</td>
</tr>
</tbody>
</table>
### Table 2-26: Measurement Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:SOUrce&lt;x&gt;</td>
<td>Sets or returns the “from” source for all single channel immediate measurements Sets or returns the source to measure “to” for two-channel measurements</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:STATE</td>
<td>Sets or returns whether the specified measurement slot is computed and displayed</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:STDdev?</td>
<td>Returns the standard deviation of values accumulated since the last statistical reset</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:TYPE</td>
<td>Sets or returns the measurement&lt;x&gt; type</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:UNIts?</td>
<td>Returns measurement&lt;x&gt; units</td>
</tr>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:VALue?</td>
<td>Returns the value of measurement&lt;x&gt;</td>
</tr>
<tr>
<td>MEASurement:METHod</td>
<td>Sets or returns the method used for calculating reference levels</td>
</tr>
<tr>
<td>MEASurement:REFLevel?</td>
<td>Returns the current reference level parameters</td>
</tr>
<tr>
<td>MEASurement:REFlevel:ABSolute:HIGH</td>
<td>Sets or returns the top reference level for rise time</td>
</tr>
<tr>
<td>MEASurement:REFlevel:ABSolute:LOW</td>
<td>Sets or returns the low reference level for rise time</td>
</tr>
<tr>
<td>MEASurement:REFlevel:ABSolute:MID</td>
<td>Sets or returns the mid reference level for measurements</td>
</tr>
<tr>
<td>MEASurement:REFlevel:ABSolute:MID&lt;x&gt;</td>
<td>Sets or returns the mid reference level in absolute units (e.g. volts)</td>
</tr>
<tr>
<td>MEASurement:REFlevel:ABSolute:MID2</td>
<td>Sets or returns the mid reference level for delay “to” measurements</td>
</tr>
<tr>
<td>MEASurement:REFlevel:METHod</td>
<td>Sets or returns the method for assigning high and low reference levels</td>
</tr>
<tr>
<td>MEASurement:REFlevel:PERCent:HIGH</td>
<td>Sets or returns the top reference percent level for rise time</td>
</tr>
<tr>
<td>MEASurement:REFlevel:PERCent:LOW</td>
<td>Sets or returns the low reference percent level for rise time</td>
</tr>
<tr>
<td>MEASurement:REFlevel:PERCent:MID</td>
<td>Sets or returns the mid reference percent level for waveform measurements</td>
</tr>
<tr>
<td>MEASurement:REFlevel:PERCent:MID2</td>
<td>Sets or returns the mid reference percent level for second waveform measurements</td>
</tr>
<tr>
<td>MEASurement:REFlevel:ABSolute:MID&lt;x&gt;</td>
<td>Sets or returns the mid reference level for the specified channel in absolute volts</td>
</tr>
<tr>
<td>MEASurement:REFlevel:PERCent:MID&lt;x&gt;</td>
<td>Sets or returns the mid reference level for the specified channel in percent</td>
</tr>
<tr>
<td>MEASurement:SNAPShot</td>
<td>Displays the measurement snapshot list</td>
</tr>
</tbody>
</table>
Table 2-26: Measurement Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASUrement:STATistics</td>
<td>Clears or returns all of the statistics accumulated for all period measurements (MEAS1 through MEAS4)</td>
</tr>
<tr>
<td>MEASUrement:STATistics:MODE</td>
<td>Turns measurement statistics on or off</td>
</tr>
<tr>
<td>MEASUrement:STATistics:WEIghting</td>
<td>Controls the responsiveness of the mean and standard deviation to waveform changes</td>
</tr>
</tbody>
</table>

### Miscellaneous Command Group

Use the commands in the Miscellaneous Command Group to perform actions that do not fit into other categories.

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

Table 2-27: Miscellaneous Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOSet</td>
<td>Sets the vertical, horizontal and trigger controls to provide a stable display of the appropriate waveform. This is equivalent to pressing the front panel Autoset button</td>
</tr>
<tr>
<td>AUTOSet:ENAble</td>
<td>Enables or disables the autoset feature</td>
</tr>
<tr>
<td>CLEARMenu</td>
<td>Clears the current menu from the display</td>
</tr>
<tr>
<td>DATE</td>
<td>Sets or returns the date displayed by the oscilloscope</td>
</tr>
<tr>
<td>*DDT</td>
<td>Sets or returns the commands that will be executed by the group execute trigger</td>
</tr>
<tr>
<td>DESkew</td>
<td>Causes the deskew values for all channels to be set to the recommended values</td>
</tr>
<tr>
<td>DESkew:DISPlay</td>
<td>Sets or returns the state of the deskew table display</td>
</tr>
<tr>
<td>FPanel:PRESS</td>
<td>Simulates the action of pressing a specified front-panel button</td>
</tr>
<tr>
<td>FPanel:TURND</td>
<td>Duplicates the action of turning a specified front-panel control knob</td>
</tr>
<tr>
<td>GPIBUsb ADDress?</td>
<td>Returns the current GPIB address</td>
</tr>
<tr>
<td>GPIBUsb:ID?</td>
<td>Returns the identification string of the connected adaptor module and firmware version</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HEADer</td>
<td>Sets or returns the Response Header Enable State</td>
</tr>
<tr>
<td>ID?</td>
<td>Returns identifying information about the oscilloscope and its firmware</td>
</tr>
<tr>
<td>*IDN?</td>
<td>Returns the same information as the ID? command except the data is formatted according to Tektronix Codes &amp; Formats</td>
</tr>
<tr>
<td>LANGUAGE</td>
<td>Sets or returns the user interface display language</td>
</tr>
<tr>
<td>LOCK</td>
<td>Sets or returns the front panel lock state</td>
</tr>
<tr>
<td>*LRN?</td>
<td>Returns a listing of oscilloscope settings</td>
</tr>
<tr>
<td>NEWpass</td>
<td>Changes the password for user protected data</td>
</tr>
<tr>
<td>PASSWord</td>
<td>Enables the *PUD and NEWpass set commands</td>
</tr>
<tr>
<td>REM</td>
<td>Specifies a comment, which is ignored by the oscilloscope</td>
</tr>
<tr>
<td>SET?</td>
<td>Returns a listing of oscilloscope settings</td>
</tr>
<tr>
<td>TEKSecure</td>
<td>Initializes both waveform and setup memories</td>
</tr>
<tr>
<td>TIME</td>
<td>Sets or returns the time displayed by the oscilloscope</td>
</tr>
<tr>
<td>TOTaluptime?</td>
<td>Returns the total number of hours that the oscilloscope has been turned on since the nonvolatile memory was last programmed</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 the status</td>
</tr>
<tr>
<td>UNLock</td>
<td>Unlocks front panel</td>
</tr>
<tr>
<td>USBTMC?</td>
<td>Returns the USBTMC information used by the USB hosts to determine the instrument interfaces</td>
</tr>
<tr>
<td>USBTMC:PRODUCTID:DECimal?</td>
<td>Returns the USBTMC product ID</td>
</tr>
<tr>
<td>USBTMC:PRODUCTID:HEXadecimal?</td>
<td>Returns the USBTMC product ID</td>
</tr>
<tr>
<td>USBTMC:SERIALnumber?</td>
<td>Returns the instrument serial number</td>
</tr>
<tr>
<td>USBTMC:VENDORID:DECimal?</td>
<td>Returns the USBTMC vendor ID</td>
</tr>
<tr>
<td>USBTMC:VENDORID:HEXadecimal?</td>
<td>Returns the USBTMC vendor ID</td>
</tr>
<tr>
<td>VERBose</td>
<td>Sets or returns the verbose state</td>
</tr>
</tbody>
</table>
Command Groups

**PictBridge Command Group**

Use the commands in the PictBridge Command Group to store printer settings.

**Table 2-28: PictBridge Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PICTBridge:DATEPrint</td>
<td>Enables or disables printing the date on the print output</td>
</tr>
<tr>
<td>PICTBridge:DEFAult</td>
<td>Sets the arguments for all PICTBridge commands to their default values</td>
</tr>
<tr>
<td>PICTBridge:IDPrint</td>
<td>Enables or disables printing the oscilloscope model and serial number on the print output</td>
</tr>
<tr>
<td>PICTBridge:IMAGESize</td>
<td>Sets or returns the image print size</td>
</tr>
<tr>
<td>PICTBridge:PAPERSize</td>
<td>Sets the output print paper size</td>
</tr>
<tr>
<td>PICTBridge:PAPERType</td>
<td>Sets or returns the paper type</td>
</tr>
<tr>
<td>PICTBridge:PRINTQual</td>
<td>Sets or returns the output print quality</td>
</tr>
<tr>
<td>USBDevice:CONFigure</td>
<td>Enables or disables the rear USB port for use with Pictbridge printers</td>
</tr>
</tbody>
</table>

**Power Command Group**

Use the commands in the Power Command Group for power analysis. The power measurements include:

- Power quality
- Switching loss
- Safe operating area
- Harmonics
- Ripple
- Modulation analysis

This command group is available when the DPO4PWR application module is installed.

**Table 2-29: Power Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWer:CURRENTSOurce</td>
<td>Sets or returns the current source for the power application</td>
</tr>
<tr>
<td>POWer:DISplay</td>
<td>Sets or returns the display state for the power application</td>
</tr>
<tr>
<td>POWer:GATESOurce</td>
<td>Sets or returns the gate source for the power application</td>
</tr>
</tbody>
</table>
Table 2-29: Power Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWer:GATING</td>
<td>Sets or returns the power application gating</td>
</tr>
<tr>
<td>PWer:HarMonics:DISPLAY:SELect</td>
<td>Sets or returns the harmonics to be displayed when the harmonics standard is None</td>
</tr>
<tr>
<td>PWer:HarMonics:DISPLAY:TYPE</td>
<td>Sets or returns the display type for harmonics tests</td>
</tr>
<tr>
<td>PWer:HarMonics:FREQRef</td>
<td>Sets or returns the frequency reference waveform for harmonics tests</td>
</tr>
<tr>
<td>PWer:HarMonics:FREQRef:FIXEDFREQValue</td>
<td>Sets or returns the fixed reference frequency value for harmonics measurements</td>
</tr>
<tr>
<td>PWer:HarMonics:IEC:CLASS</td>
<td>Sets or returns the filtering class for IEC harmonics</td>
</tr>
<tr>
<td>PWer:HarMonics:IEC:FILTER</td>
<td>Sets or returns the enabled state for filtering of IEC harmonics</td>
</tr>
<tr>
<td>PWer:HarMonics:IEC:FUNDamental</td>
<td>Sets or returns the fundamental current for IEC harmonics</td>
</tr>
<tr>
<td>PWer:HarMonics:IEC:GROUPing</td>
<td>Sets or returns the enabled state for grouping of IEC harmonics</td>
</tr>
<tr>
<td>PWer:HarMonics:IEC:INPUTPOWER</td>
<td>Sets of returns the class D input power for IEC harmonics</td>
</tr>
<tr>
<td>PWer:HarMonics:IEC:LINEFREQUENCY</td>
<td>Sets or returns the line frequency for the IEC standard</td>
</tr>
<tr>
<td>PWer:HarMonics:IEC:OBSPERIOD</td>
<td>Sets or returns the IEC observation period</td>
</tr>
<tr>
<td>PWer:HarMonics:IEC:POWERFACTOR</td>
<td>Sets or returns the power factor for IEC harmonics</td>
</tr>
<tr>
<td>PWer:HarMonics:MIL:FUNDamental:CALCmethod</td>
<td>Sets or returns the measurement method for the MIL harmonics fundamental frequency</td>
</tr>
<tr>
<td>PWer:HarMonics:MIL:LINEFREQUENCY</td>
<td>Sets or returns the line frequency for MIL-STD-1399 harmonics tests</td>
</tr>
<tr>
<td>PWer:HarMonics:MIL:POWERLEVEL</td>
<td>Sets or returns the power level for MIL-STD-1399 harmonics tests</td>
</tr>
<tr>
<td>PWer:HarMonics:NR_HARMonics</td>
<td>Sets of returns the number of harmonics (a value in the range of 20 to 400) when the harmonics standard is NONE</td>
</tr>
<tr>
<td>PWer:HarMonics:RESults:HAR&lt;1-400&gt;:FREQuency?</td>
<td>Returns the frequency of the harmonic</td>
</tr>
</tbody>
</table>
Table 2-29: Power Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWer:HARMonics:RESults:HAR&lt;1-400&gt;:IECMAX?</td>
<td>The IEC standard specifies harmonics measurements to be computed in windows of time, with each time window being nominally 200 ms. This returns the maximum of the RMS magnitude of the harmonic, computed across successive 200 ms time windows within an observation period entered by the user.</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:HAR&lt;1-400&gt;:LIMIT?</td>
<td>The IEC and MIL standards specify a limit for each harmonic magnitude. Returns the limit in absolute units, or as a percentage of the fundamental as specified by the standard. IEC Class C (Table 2) and MIL standards specify the limit as a percentage of the fundamental.</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:HAR&lt;1-400&gt;:PHASE?</td>
<td>Returns the phase of the harmonic in degrees. The phase is measured relative to the zero-crossing of the reference waveform. When there is no reference waveform, the phase is relative to the fundamental component.</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:HAR&lt;1-400&gt;:RMS:ABSolute?</td>
<td>Returns the RMS magnitude of the harmonic expressed in absolute units.</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:HAR&lt;1-400&gt;:RMS:PERCent?</td>
<td>Returns the RMS magnitude of the harmonic expressed as a percentage of the fundamental.</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:HAR&lt;1-400&gt;:TEST:IEC:CLASSALIMIT?</td>
<td>Returns PASS, FAIL or NA. Specifies if the IEC Class A higher harmonic limit (and conditions) are met.</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:HAR&lt;1-400&gt;:TEST:IEC:NORMAL?</td>
<td>Returns PASS, FAIL or NA. Specifies if the Normal IEC harmonic limits are met.</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:HAR&lt;1-400&gt;:TEST:IEC:POHCLIMIT?</td>
<td>Returns PASS, FAIL or NA. Specifies if the higher harmonic limit (and conditions) for the 21st and higher order odd harmonics are met.</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:IEC:HARM3ALTernate?</td>
<td>Returns the IEC harmonics test result for the 3rd harmonic: PASS, FAIL or NA.</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:IEC:HARM5ALTernate?</td>
<td>Returns the IEC harmonics test result for the 5th harmonic: PASS, FAIL or NA.</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:IEC:POHC?</td>
<td>Returns the IEC POHC measurement.</td>
</tr>
</tbody>
</table>
### Table 2-29: Power Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWer:HARMonics:RESults:IEC:POHL?</td>
<td>Returns the IEC POHL measurement</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:IEC:POWer?</td>
<td>Returns the IEC input power measurement</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:IEC:POWERFactor?</td>
<td>Returns the IEC power factor measurement</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:PASSFail?</td>
<td>Returns the overall harmonics test result: PASS, FAIL or NA</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:RMS?</td>
<td>Returns the root mean square value of the source waveform</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:SAVE</td>
<td>Saves the harmonic results to the specified file in CSV format</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:THDF?</td>
<td>Returns the Total Harmonic Distortion (THD) in percentage, measured as a ratio to the RMS value of the fundamental component of the source waveform</td>
</tr>
<tr>
<td>POWer:HARMonics:RESults:THDR?</td>
<td>Returns the Total Harmonic Distortion (THD) in percentage, measured as a ratio to the RMS value of the source waveform</td>
</tr>
<tr>
<td>POWer:HARMonics:SOURce</td>
<td>Sets or returns the source waveform for harmonics tests</td>
</tr>
<tr>
<td>POWer:HARMonics:STANDard</td>
<td>Sets or returns the standard for harmonics tests</td>
</tr>
<tr>
<td>POWer:INDICators</td>
<td>Sets or returns the state of the measurement indicators for the power application</td>
</tr>
<tr>
<td>POWer:MODulation:SOUrce</td>
<td>Sets or returns the source waveform for modulation tests</td>
</tr>
<tr>
<td>POWer:MODulation:TYPe</td>
<td>Sets or returns the modulation type</td>
</tr>
<tr>
<td>POWer:QUALity:APPpwr?</td>
<td>Returns the apparent power measurement</td>
</tr>
<tr>
<td>POWer:QUALity:DISplay:APPpwr</td>
<td>Sets or returns the display state for the apparent power readout</td>
</tr>
<tr>
<td>POWer:QUALity:DISplay:FREQuency</td>
<td>Sets or returns the display state for the frequency readout</td>
</tr>
<tr>
<td>POWer:QUALity:DISplay:ICRESTfactor</td>
<td>Sets or returns the display state for the current crest factor readout</td>
</tr>
<tr>
<td>POWer:QUALity:DISplay:IRMS</td>
<td>Sets or returns the display state for the rms current (IRMS) readout</td>
</tr>
<tr>
<td>POWer:QUALity:DISplay:PHASEangle</td>
<td>Sets or returns the display state for the phase angle readout</td>
</tr>
<tr>
<td>POWer:QUALity:DISplay:POWERFACTor</td>
<td>Sets or returns the display state for the power factor readout</td>
</tr>
<tr>
<td>POWer:QUALity:DISplay:REACTpwr</td>
<td>Sets or returns the display state for the reactive power readout</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>POWer:QUALity:DISplay:TRUEpwr</td>
<td>Sets or returns the display state for the true power readout</td>
</tr>
<tr>
<td>POWer:QUALity:DISplay:VCRESTfactor</td>
<td>Sets or returns the display state for the voltage crest factor readout</td>
</tr>
<tr>
<td>POWer:QUALity:DISplay:VRMS</td>
<td>Sets or returns the display state for the rms voltage (VRMS) readout</td>
</tr>
<tr>
<td>POWer:QUALity:FREQREference</td>
<td>Sets or returns the power quality frequency reference</td>
</tr>
<tr>
<td>POWer:QUALity:FREQuency?</td>
<td>Returns the frequency measurement</td>
</tr>
<tr>
<td>POWer:QUALity:ICRESTfactor?</td>
<td>Returns the current crest factor measurement</td>
</tr>
<tr>
<td>POWer:QUALity:IRMS?</td>
<td>Returns the rms current measurement</td>
</tr>
<tr>
<td>POWer:QUALity:PHASEangle?</td>
<td>Returns the phase angle measurement</td>
</tr>
<tr>
<td>POWer:QUALity:POWERAFactor?</td>
<td>Returns the power factor measurement</td>
</tr>
<tr>
<td>POWer:QUALity:REACTpwr?</td>
<td>Returns the reactive power measurement</td>
</tr>
<tr>
<td>POWer:QUALity:TRUEpwr?</td>
<td>Returns the true power measurement</td>
</tr>
<tr>
<td>POWer:QUALity:VRMS?</td>
<td>Returns the rms voltage measurement</td>
</tr>
<tr>
<td>POWer:REFLevel:ABSolute</td>
<td>Sets the reference levels to their default unit values</td>
</tr>
<tr>
<td>POWer:REFLevel:ABSolute:HIGH</td>
<td>Sets or returns the top reference level for rise time</td>
</tr>
<tr>
<td>POWer:REFLevel:ABSolute:LOW</td>
<td>Sets or returns the low reference level for rise time</td>
</tr>
<tr>
<td>POWer:REFLevel:ABSolute:MID&lt;1-3&gt;</td>
<td>Sets or returns the mid reference level for measurements</td>
</tr>
<tr>
<td>POWer:REFLevel:HYSTeresis</td>
<td>Sets or returns the measurement reference level hysteresis value</td>
</tr>
<tr>
<td>POWer:REFLevel:METHod</td>
<td>Sets or returns the method used to calculate the 0% and 100% reference level</td>
</tr>
<tr>
<td>POWer:REFLevel:PERCent</td>
<td>Sets the reference levels to the default percentage values</td>
</tr>
<tr>
<td>POWer:REFLevel:PERCent:HIGH</td>
<td>Sets or returns the top reference percent level for rise time</td>
</tr>
<tr>
<td>POWer:REFLevel:PERCent:LOW</td>
<td>Sets or returns the low reference percent level for rise time</td>
</tr>
<tr>
<td>POWer:REFLevel:PERCent:MID&lt;1-3&gt;</td>
<td>Sets or returns the mid reference percent level for waveform measurements</td>
</tr>
<tr>
<td>POWer:RIPPLE</td>
<td>Sets the vertical offset of the source waveform</td>
</tr>
<tr>
<td>POWer:RIPPLE:RESults:AMPLitude?</td>
<td>Returns the peak-to-peak ripple measurement</td>
</tr>
</tbody>
</table>
Table 2-29: Power Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWer:RIPplE:RESults:MAX?</td>
<td>Returns the maximum of the peak-to-peak ripple measurements</td>
</tr>
<tr>
<td>POWer:RIPplE:RESults:MEAN?</td>
<td>Returns the mean of the peak-to-peak ripple measurements</td>
</tr>
<tr>
<td>POWer:RIPplE:RESults:MIN?</td>
<td>Returns the minimum of the peak-to-peak ripple measurement</td>
</tr>
<tr>
<td>POWer:RIPplE:RESults:STDdev?</td>
<td>Returns the standard deviation of the peak-to-peak ripple measurements</td>
</tr>
<tr>
<td>POWer:RIPplE:SOUrce</td>
<td>Sets or returns the source waveform for ripple tests</td>
</tr>
<tr>
<td>POWer:SOA:LINear:XMAX</td>
<td>Sets or returns the user XMAX value for use in linear SOA calculations</td>
</tr>
<tr>
<td>POWer:SOA:LINear:XMIN</td>
<td>Sets or returns the user XMIN value for use in linear SOA calculations</td>
</tr>
<tr>
<td>POWer:SOA:LINear:YMAX</td>
<td>Sets or returns the user YMAX value for use in linear SOA calculations</td>
</tr>
<tr>
<td>POWer:SOA:LINear:YMIN</td>
<td>Sets or returns the user YMIN value for use in linear SOA calculations</td>
</tr>
<tr>
<td>POWer:SOA:LOG:XMAX</td>
<td>Sets or returns the user XMAX value for use in log SOA calculations</td>
</tr>
<tr>
<td>POWer:SOA:LOG:XMIN</td>
<td>Sets or returns the user XMIN value for use in log SOA calculations</td>
</tr>
<tr>
<td>POWer:SOA:LOG:YMAX</td>
<td>Sets or returns the user YMAX value for use in log SOA calculations</td>
</tr>
<tr>
<td>POWer:SOA:LOG:YMIN</td>
<td>Sets or returns the user YMIN value for use in log SOA calculations</td>
</tr>
<tr>
<td>POWer:SOA:MASK:DEFine</td>
<td>Sets or returns the X (Volts) and Y (Amps) coordinates of the current SOA mask</td>
</tr>
<tr>
<td>POWer:SOA:MASK:MAXAmps</td>
<td>Sets or returns the maximum current applied to SOA mask testing</td>
</tr>
<tr>
<td>POWer:SOA:MASK:MAXVolts</td>
<td>Sets or returns the maximum voltage applied to SOA mask testing</td>
</tr>
<tr>
<td>POWer:SOA:MASK:MAXWatts</td>
<td>Sets or returns the maximum power applied to SOA mask testing</td>
</tr>
<tr>
<td>POWer:SOA:MASK:NR_PT?</td>
<td>Returns the number of mask points defined</td>
</tr>
<tr>
<td>POWer:SOA:MASK:STATe</td>
<td>Sets or returns the state of the mask for SOA calculations</td>
</tr>
<tr>
<td>POWer:SOA:MASK:STOPOnviol</td>
<td>Sets or returns the enabled state of the mask stop on violation condition</td>
</tr>
<tr>
<td>POWer:SOA:PLOTTYPe</td>
<td>Sets or returns the SOA plot type</td>
</tr>
<tr>
<td>POWer:SOA:RESults:FAILures:QTY?</td>
<td>Returns the number of failures in the test</td>
</tr>
</tbody>
</table>
### Table 2-29: Power Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>POWer:SOA:RESult:NUMACq?</code></td>
<td>Returns the number of acquisitions in the test</td>
</tr>
<tr>
<td><code>POWer:SOA:RESult:STATe?</code></td>
<td>Returns the pass/fail state of the SOA test</td>
</tr>
<tr>
<td><code>POWer:STATistics</code></td>
<td>Clears all the accumulated statistics of all measurements</td>
</tr>
<tr>
<td><code>POWer:STATistics:MODE</code></td>
<td>Enables or disables the display of the measurement statistics</td>
</tr>
<tr>
<td><code>POWer:STATistics:WEIghting</code></td>
<td>Sets the number of samples which are included for the statistics computations for mean and the standard deviation</td>
</tr>
<tr>
<td><code>POWer:SWLoss:CONDCALCmethod</code></td>
<td>Sets or returns the power application switching loss conduction calculation method</td>
</tr>
<tr>
<td><code>POWer:SWLoss:CONDuction:ENERGY:MAX?</code></td>
<td>Returns the maximum conduction energy for the switching loss calculation</td>
</tr>
<tr>
<td><code>POWer:SWLoss:CONDuction:ENERGY:MEAN?</code></td>
<td>Returns the mean conduction energy for the switching loss calculation</td>
</tr>
<tr>
<td><code>POWer:SWLoss:CONDuction:ENERGY:MIN?</code></td>
<td>Returns the minimum conduction energy for the switching loss calculation</td>
</tr>
<tr>
<td><code>POWer:SWLoss:CONDuction:POWER:MAX?</code></td>
<td>Returns the maximum conduction power for the switching loss calculation</td>
</tr>
<tr>
<td><code>POWer:SWLoss:CONDuction:POWER:MEAN?</code></td>
<td>Returns the mean conduction power for the switching loss calculation</td>
</tr>
<tr>
<td><code>POWer:SWLoss:CONDuction:POWER:MIN?</code></td>
<td>Returns the minimum conduction power for the switching loss calculation</td>
</tr>
<tr>
<td><code>POWer:SWLoss:DISplay</code></td>
<td>Sets or returns the display selection for switching loss results</td>
</tr>
<tr>
<td><code>POWer:SWLoss:GATe:POLarity</code></td>
<td>Sets or returns the switching loss gate polarity</td>
</tr>
<tr>
<td><code>POWer:SWLoss:GATe:TURNON</code></td>
<td>Sets or returns the gate turn on level for switching loss power measurements</td>
</tr>
<tr>
<td><code>POWer:SWLoss:NUMCYCles?</code></td>
<td>Returns the number of cycles counted for the switching loss calculation</td>
</tr>
<tr>
<td><code>POWer:SWLoss:RDSon</code></td>
<td>Sets or returns RDSON value for use in switching loss calculations when the conduction calculation method is RDSON</td>
</tr>
<tr>
<td><code>POWer:SWLoss:REFLevel:ABSolute:GATEMid</code></td>
<td>Sets or returns the mid voltage reference level used in switching loss power measurements in volts</td>
</tr>
<tr>
<td><code>POWer:SWLoss:REFLevel:ABSolute:LOWCurrent</code></td>
<td>Sets or returns the low current reference level used in switching loss power measurements in amperes</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>POWer:SWLoss:REFLevel:ABSolute:LOWVoltage</td>
<td>Sets or returns the low voltage reference level used in switching loss power measurements in volts</td>
</tr>
<tr>
<td>POWer:SWLoss:REFLevel:PERCent:GATEMid</td>
<td>Sets or returns the mid voltage reference level used in switching loss power measurements in percentage</td>
</tr>
<tr>
<td>POWer:SWLoss:REFLevel:PERCent:LOWCurrent</td>
<td>Sets or returns the low current reference level used in switching loss power measurements in percentage</td>
</tr>
<tr>
<td>POWer:SWLoss:REFLevel:PERCent:LOWVoltage</td>
<td>Sets or returns the low voltage reference level used in switching loss power measurements in percentage</td>
</tr>
<tr>
<td>POWer:SWLoss:TOFF:ENERGY:MAX?</td>
<td>Returns the maximum Toff energy switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TOFF:ENERGY:MEAN?</td>
<td>Returns the mean Toff energy switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TOFF:ENERGY:MIN?</td>
<td>Returns the minimum Toff energy switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TOFF:POWER:MAX?</td>
<td>Returns the maximum Toff power switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TOFF:POWER:MEAN?</td>
<td>Returns the mean Toff power switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TOFF:POWER:MIN?</td>
<td>Returns the minimum Toff power switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TON:ENERGY:MAX?</td>
<td>Returns the maximum Ton energy switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TON:ENERGY:MEAN?</td>
<td>Returns the mean Ton energy switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TON:ENERGY:MIN?</td>
<td>Returns the minimum Ton energy switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TON:POWER:MAX?</td>
<td>Returns the maximum Ton power switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TON:POWER:MEAN?</td>
<td>Returns the mean Ton power switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TON:POWER:MIN?</td>
<td>Returns the minimum Ton power switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TOTal:ENERGY:MAX?</td>
<td>Returns the maximum total energy switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TOTal:ENERGY:MEAN?</td>
<td>Returns the mean total energy switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TOTal:ENERGY:MIN?</td>
<td>Returns the minimum total energy switching loss calculation</td>
</tr>
</tbody>
</table>
Table 2-29: Power Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWer:SWLoss:TOTal:POWER:MAX?</td>
<td>Returns the maximum total power switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TOTal:POWER:MEAN?</td>
<td>Returns the mean total power switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:TOTal:POWER:MIN?</td>
<td>Returns the minimum total power switching loss calculation</td>
</tr>
<tr>
<td>POWer:SWLoss:VCEsat</td>
<td>Sets or returns VCESAT value for use in switching loss calculations when the conduction calculation method is VCESAT</td>
</tr>
<tr>
<td>POWer:TYPe</td>
<td>Sets or returns the power application measurement type</td>
</tr>
<tr>
<td>POWer:VOLTAGE:SOURCE</td>
<td>Sets or returns the voltage source for the power application</td>
</tr>
</tbody>
</table>

Save and Recall Command Group

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

Table 2-30: Save and Recall Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FACTory</td>
<td>Resets the oscilloscope to factory default settings</td>
</tr>
<tr>
<td>*RCL</td>
<td>Recalls saved oscilloscope settings</td>
</tr>
<tr>
<td>RECALL:SETUp</td>
<td>Recalls saved oscilloscope settings</td>
</tr>
<tr>
<td>RECALL:WAVEform</td>
<td>Recalls a stored waveform to a reference location</td>
</tr>
<tr>
<td>*SAV</td>
<td>Stores the state of the oscilloscope to a specified memory location</td>
</tr>
<tr>
<td>SAVE:ASSign:TYPe</td>
<td>Sets or returns the assignment of the save button</td>
</tr>
<tr>
<td>SAVE:EVENTtable:BUS&lt;x&gt;</td>
<td>Saves event table data from bus&lt;x&gt; to a specified file</td>
</tr>
<tr>
<td>SAVE:IMAGE</td>
<td>Saves a capture of the screen image to the specified file</td>
</tr>
</tbody>
</table>
Table 2-30: Save and Recall Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVE:IMAGE:FILEFormat</td>
<td>Sets or returns the file format to use for saving screen images. The file format is not automatically determined by the file name extension. You need to choose a file name with an extension which is consistent with the selected file format.</td>
</tr>
<tr>
<td>SAVE:IMAGE:INKSaver</td>
<td>Sets or returns the current inksaver setting for the SAVE:IMAGE command.</td>
</tr>
<tr>
<td>SAVE:IMAGE:LAYout</td>
<td>Sets or returns the layout to use for saved screen images.</td>
</tr>
<tr>
<td>SAVE:SETUp</td>
<td>Saves the state of the oscilloscope to a specified memory location or file.</td>
</tr>
<tr>
<td>SAVE:WAVEform</td>
<td>Saves a waveform to one of the reference memory locations or a file.</td>
</tr>
<tr>
<td>SAVE:WAVEform:FILEFormat</td>
<td>Sets or returns the format for saved waveforms.</td>
</tr>
<tr>
<td>SAVE:WAVEform:GATING</td>
<td>Specifies whether save waveform operations should save the entire waveform or a specified portion of the waveform.</td>
</tr>
</tbody>
</table>

Search Command Group

Use the commands in the Search Commands Group to seek out information in waveform records.

Search Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH?</td>
<td>Returns all search-related settings</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:COPY</td>
<td>Copies the search criteria to the trigger, or the trigger criteria to the search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:STATE</td>
<td>Sets the search state to on or off</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TOTAL?</td>
<td>Returns the total number of matches for search &lt;x&gt;</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS?</td>
<td>Returns the search trigger bus type</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:B&lt;x&gt;:AUDio:CONDition</td>
<td>Sets or returns the search trigger condition for the AUDIO bus: Start of Frame or Data</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:AUDio:DATA:HIVALue</td>
<td>Sets or returns the search trigger data upper word for the AUDIO bus</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:AUDio:DATA:OFFSET</td>
<td>Sets or returns the search trigger data offset for the AUDIO bus</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:AUDio:DATA:QUALifier</td>
<td>Sets or returns the search trigger data qualifier for the AUDIO bus</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:AUDio:DATA:VALue</td>
<td>Sets or returns the search trigger data lower word for the AUDIO bus</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:AUDio:WORD</td>
<td>Sets or returns the search trigger data alignment for the AUDIO bus</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:CAN:CONDition</td>
<td>Sets or returns the search condition for CAN search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:CAN:DATA:DIRection</td>
<td>Sets or returns the CAN search condition to be valid on a READ, WRITE or either</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:CAN:DATA:QUALifier</td>
<td>Sets or returns the CAN data qualifier</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:CAN:DATA:SIZE</td>
<td>Sets or returns the length of the data string in bytes to be used for CAN search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:CAN:DATA:VALue</td>
<td>Sets or returns the binary data string to be used for CAN search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:CAN:FRAMEtype</td>
<td>Sets or returns the CAN Frame Type to be used</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:CAN:(Identifier</td>
<td>Address):MODe</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:CAN:(Identifier</td>
<td>Address):VALue</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:CONDition</td>
<td>Sets or returns the trigger condition for FLEXRAY</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:CYCLEcount:HIVALue</td>
<td>Sets or returns the binary data string to be used for FLEXRAY cycle count high value</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:CYCLEcount:QUALifier</td>
<td>Sets or returns the FLEXRAY cycle count qualifier</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:CYCLEcount:VALue</td>
<td>Sets or returns the binary data string to be used for FLEXRAY cycle count low value</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:DATA:HIVALue</td>
<td>Sets or returns the high binary data string</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:DATA:OFFSET</td>
<td>Sets or returns the offset of the data string in bytes</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:DATA:QUALifier</td>
<td>Sets or returns the FLEXRAY data qualifier</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:B&lt;x&gt;:FLEXray:DATA:SIZE</td>
<td>Sets or returns the length of the data string in bytes</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:DATA:VALUE</td>
<td>Sets or returns the low binary data string</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:EOFFTYPE</td>
<td>Sets or returns the end of frame type</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:ERTYPE</td>
<td>Sets or returns the error type</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:FRAMEID:HIVALue</td>
<td>Sets or returns the binary data string used for FLEXRAY frame ID high value</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:FRAMEID:QUALifier</td>
<td>Sets or returns the FLEXRAY frame ID qualifier</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:FRAMEID:VALUE</td>
<td>Sets or returns the binary data string to be used for FLEXRAY frame ID low value</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:FRAMEType</td>
<td>Sets or returns the frame type</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:HEADER:CRC</td>
<td>Sets or returns the CRC portion of the binary header string</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:HEADER:CYCLEcount</td>
<td>Sets or returns the cycle count portion of the binary header string</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:HEADER:FRAMEID</td>
<td>Sets or returns the frame id portion of the binary header string</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:HEADER:INDBits</td>
<td>Sets or returns the indicator bits portion of the binary header string</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:FLEXray:HEADER:PAYLength</td>
<td>Sets or returns the payload length portion of the binary header string</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:I2C:ADDRes:MODE</td>
<td>Sets or returns the I2C address mode to 7 or 10-Bit</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:I2C:ADDRes:TYPE</td>
<td>Sets or returns the I2C address type to I2C special addresses</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:I2C:ADDRes:VALUE</td>
<td>Sets or returns the binary address string to be used for I2C search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:I2C:CONDition</td>
<td>Sets or returns the search condition for I2C search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:I2C:DATA:Direction</td>
<td>Sets or returns the I2C search condition to be valid on a READ, WRITE or either</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:I2C:DATA:SIZE</td>
<td>Sets or returns the length of the data string in bytes to be used for I2C search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:I2C:DATA:VALUE</td>
<td>Sets or returns the binary data string to be used for I2C search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:LIN:CONDition</td>
<td>Sets or returns the search condition for a LIN search</td>
</tr>
</tbody>
</table>
### Search Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:LIN:DATA:HIVALue}</td>
<td>Sets or returns the binary data string</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:LIN:DATA:QUALifier}</td>
<td>Sets or returns the LIN data qualifier</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:LIN:DATA:SIZE}</td>
<td>Sets or returns the length of the data string in bytes</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:LIN:DATA:VALUE}</td>
<td>Sets or returns the binary data string used for a LIN search</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:LIN:ERROR}</td>
<td>Sets or returns the error type used for a LIN search</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:LIN:IDENTIFIER:VALUE}</td>
<td>Sets or returns the binary address string used for LIN search</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:PARALLEL:VALUE}</td>
<td>Sets or returns the binary data string to be used for a Parallel search</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:RS232C:CONDITION}</td>
<td>Sets or returns the trigger condition for an RS-232 trigger</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:RS232C:RX:DATA:SIZE}</td>
<td>Sets or returns the length of the data string for an RS-232 trigger, if the trigger condition is RX</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:RS232C:RX:DATA:VALUE}</td>
<td>Sets or returns the binary data string for an RS-232 trigger, if the condition involves RX</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:RS232C:TX:DATA:SIZE}</td>
<td>Sets or returns the length of the data string to be used for an RS-232 Trigger, if the Trigger condition is TX</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:RS232C:TX:DATA:VALUE}</td>
<td>Sets or returns the binary data string to be used for an RS-232 trigger, if the condition involves RX</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:SPI:CONDITION}</td>
<td>Sets or returns the search condition for SPI search</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:SPI:DATA:{MISO</td>
<td>IN}:VALUE}</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:SPI:DATA:{MOSI</td>
<td>OUT}:VALUE}</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:SPI:DATA:SIZE}</td>
<td>Sets or returns the length of the data string in bytes to be used for SPI search</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:USB:ADDRESS:HIVALue}</td>
<td>Sets or returns the high limit for USB address searches</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:USB:ADDRESS:VALUE}</td>
<td>Sets or returns the value for USB address searches</td>
</tr>
<tr>
<td>\texttt{SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:USB:CONDITION}</td>
<td>Sets or returns the USB search trigger condition</td>
</tr>
</tbody>
</table>
### Search Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:&lt;BR&gt;:USB:DATa:HIVALue</td>
<td>Sets or returns the high limit for USB data searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:&lt;BR&gt;:USB:DATa:OFFSET</td>
<td>Sets or returns the data offset for USB data searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:&lt;BR&gt;:USB:DATa:SIZE</td>
<td>Sets or returns the number of data bytes for USB searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:&lt;BR&gt;:USB:DATa:TYPe</td>
<td>Sets or returns the data type for USB searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:&lt;BR&gt;:USB:DATa:VALue</td>
<td>Sets or returns the data value for USB data searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:&lt;BR&gt;:USB:ENDPoint:VALue</td>
<td>Sets or returns the endpoint value for USB searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:&lt;BR&gt;:USB:ERRORType</td>
<td>Sets or returns the error type for USB searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:&lt;BR&gt;:USB:HANDSHAKEType</td>
<td>Sets or returns the handshake type for USB searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:&lt;BR&gt;:USB:QUALifier</td>
<td>Sets or returns the qualifier for USB searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:&lt;BR&gt;:USB:SOFFRAME:NUMber</td>
<td>Sets or returns the SOF number for USB searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:&lt;BR&gt;:USB:SPECIALType</td>
<td>Sets or returns the special packet type for USB searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:&lt;BR&gt;:USB:TOKENType</td>
<td>Sets or returns the token type for USB searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:&lt;BR&gt;:SOUrce</td>
<td>Sets or returns the bus for a serial search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:EDGE:SLOpe</td>
<td>Sets or returns the slope for an edge search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:EDGE:SOUrce</td>
<td>Sets or returns the source waveform for an edge search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LEVel</td>
<td>Sets or returns the level for an edge search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LEVel:CH&lt;x&gt;</td>
<td>Sets or returns the level for an edge search of the specified channel</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LEVel:MATH</td>
<td>Sets or returns the math waveform level for edge search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LEVel:REF&lt;x&gt;</td>
<td>Sets or returns the reference waveform level for edge search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:FUNCtion</td>
<td>Sets or returns the logic operator for the logic search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGIc:INPut:CH&lt;x&gt;</td>
<td>Sets or returns the Boolean logic criteria for the logic search</td>
</tr>
</tbody>
</table>
## Search Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:INPut:CLOCk:EDGE</td>
<td>Sets or returns whether the clock edge is rise or fall for a logic search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:INPut:CLOCk:SOURCE</td>
<td>Sets or returns the clock source definition for logic search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:INPut:D&lt;x&gt;</td>
<td>Sets or returns the criteria for a logic search to determine where to place a mark for digital channel &lt;x&gt;</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:INPut:MATH</td>
<td>Sets or returns the Boolean logic criteria for the logic search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:INPut:REF&lt;x&gt;</td>
<td>Sets or returns the Boolean logic criteria for the logic search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:PATtern:INPut:CH&lt;x&gt;</td>
<td>Sets or returns the criteria for a pattern search to determine where to place a mark for digital channel &lt;x&gt;</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:PATtern:INPut:D&lt;x&gt;</td>
<td>Sets or returns the Boolean logic criteria for the logic search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:PATtern:INPut:MATH</td>
<td>Sets or returns the Boolean logic criteria for the logic search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:PATtern:INPut:REF&lt;x&gt;</td>
<td>Sets or returns the Boolean logic criteria for the logic search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:WHen</td>
<td>Sets or returns the condition for generating a logic pattern search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:WHen:LESSLimit</td>
<td>Sets or returns the maximum time that the selected pattern may be true</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:WHen:MORELimit</td>
<td>Sets or returns the minimum time that the selected pattern may be true</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:THresh:CH&lt;x&gt;</td>
<td>Sets or returns the channel threshold level for an logic search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:THresh:MATH</td>
<td>Sets or returns the math waveform threshold level for logic search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:THresh:REF&lt;x&gt;</td>
<td>Sets or returns the reference waveform threshold level for logic search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOWer:CH&lt;x&gt;</td>
<td>Sets or returns the lower waveform threshold level for all channel waveform searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOWer:MATH</td>
<td>Sets or returns the lower waveform threshold level for all math waveform searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOWer:REF&lt;x&gt;</td>
<td>Sets or returns the lower waveform threshold level for all reference waveform searches</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:PULSE:POLarity</td>
<td>Sets or returns the polarity for a pulse search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:PULSE:SOURCE</td>
<td>Sets or returns the source waveform for a pulse search</td>
</tr>
</tbody>
</table>
## Search Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:PULSEWidth:WHen</td>
<td>Sets or returns the condition for generating a pulse width search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:PULSEWidth:Width</td>
<td>Sets or returns the pulse width setting for a pulse width search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:RUNT:POLarity</td>
<td>Sets or returns the polarity setting for a runt search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:RUNT:SOUrce</td>
<td>Sets or returns the source setting for a runt search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:RUNT:WHen</td>
<td>Sets or returns the condition setting for a runt search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:RUNT:Width</td>
<td>Sets or returns the width setting for a runt search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SETHold:CLOCk:EDGE</td>
<td>Sets or returns the clock slope setting for a setup/hold search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SETHold:CLOCk:SOUrce</td>
<td>Sets or returns the clock source setting for an setup/hold search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SETHold:CLOCk:THReshold</td>
<td>Sets or returns the clock threshold setting for an setup/hold search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SETHold:DATa:SOUrce</td>
<td>Sets or returns the data source setting for an setup/hold search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SETHold:DATa:THReshold</td>
<td>Sets or returns the data threshold setting for an setup/hold search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SETHold:HOLDTime</td>
<td>Sets or returns the hold time setting for an setup/hold search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SETHold:SETTime</td>
<td>Sets or returns the setup time setting for an setup/hold search</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SETHold:THReshold:CH&lt;x&gt;</td>
<td>Sets or returns the trigger search setup and hold threshold for the selected channel</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SETHold:THReshold:{MATH</td>
<td>MATH1}</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:SETHold:THReshold:REF&lt;x&gt;</td>
<td>Sets or returns the trigger search setup and hold threshold for the selected reference waveform</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:{TRANSition</td>
<td>RISEFall}:DELTatime</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:{TRANSition</td>
<td>RISEFall}:POLarity</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:{TRANSition</td>
<td>RISEFall}:SOUrce</td>
</tr>
<tr>
<td>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:{TRANSition</td>
<td>RISEFall}:WHen</td>
</tr>
</tbody>
</table>
Search Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;:TRIGger:A:TYPE</td>
<td>Sets or returns the trigger type setting for a search</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;:TRIGger:A:UPPerthreshold:CH&lt;x&gt;</td>
<td>Sets or returns the waveform upper threshold level for all channel waveform searches</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;:TRIGger:A:UPPerthreshold:MATH</td>
<td>Sets or returns the waveform upper threshold level for all math waveform searches</td>
</tr>
<tr>
<td>SEARCH: SEARCH&lt;x&gt;:TRIGger:A:UPPerthreshold:REF&lt;x&gt;</td>
<td>Sets or returns the waveform upper threshold level for all reference waveform searches</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 oscilloscope and control events.

Several commands and queries used with the oscilloscope are common to all IEEE488.2 compliant devices. The IEEE Std 488.2-1987 defines these commands and queries. The common commands begin with an asterisk (*) character.

Table 2-31: 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 oscilloscope status</td>
</tr>
<tr>
<td>*CLS</td>
<td>Clears status</td>
</tr>
<tr>
<td>DESE</td>
<td>Sets or returns the bits in the Device Event Status Enable Register</td>
</tr>
<tr>
<td>*ESE</td>
<td>Sets or returns 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>Return number of events in the event 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 &quot;1&quot; when all current operations are finished</td>
</tr>
<tr>
<td>*PSC</td>
<td>Sets or returns the power on status flag</td>
</tr>
<tr>
<td>*PUD</td>
<td>Sets or returns a string of protected user data</td>
</tr>
</tbody>
</table>
Table 2-31: Status and Error Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*RST</td>
<td>Resets the oscilloscope to factory default settings</td>
</tr>
<tr>
<td>*SRE</td>
<td>Sets or returns 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 oscilloscope from executing further commands until all pending operations finish</td>
</tr>
</tbody>
</table>

Trigger Command Group

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

There are two triggers: A and B. Where appropriate, this command set has parallel construction between triggers.

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 triggers to pulse, logic, or video modes. With pulse triggering, the oscilloscope triggers whenever it detects a pulse of a certain width or height. Logic triggering lets you logically combine the signals on one or channels. The oscilloscope then triggers when it detects a certain combination of signal levels. Video triggering enables you to trigger on the most common Standard Definition video standards.

Table 2-32: 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</td>
</tr>
<tr>
<td>TRIGger:A</td>
<td>Sets A trigger level to 50% or returns current A trigger parameters</td>
</tr>
<tr>
<td>TRIGger:A:BUS</td>
<td>Sets or returns the serial trigger type</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:AUDio:CONDition</td>
<td>Sets or returns the trigger condition for the AUDIO bus: Start of Frame or Data</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:AUDio:DATa:HIValue</td>
<td>Sets or returns the trigger data upper word for the AUDIO bus</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:AUDio:DATa:OFFSet</td>
<td>Sets or returns the trigger data offset for the AUDIO bus</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:AUDio:DATa:QUALifier</td>
<td>Sets or returns the trigger data qualifier for the AUDIO bus</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:AUDio:DATa:VALue</td>
<td>Sets or returns the trigger data lower word for the AUDIO bus</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:AUDio:DATa:WORD</td>
<td>Sets or returns the trigger data alignment for the AUDIO bus.</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:CAN:CONDITION</td>
<td>Sets or returns the CAN condition</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:CAN:DATa:DIRection</td>
<td>Sets or returns the CAN trigger condition to be valid on a READ, WRITE, or either</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:CAN:DATa:QUALifier</td>
<td>Sets or returns the CAN data qualifier</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:CAN:DATa:SIZE</td>
<td>Sets or returns the length of the data string in bytes to be used for CAN trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:CAN:DATa:VALue</td>
<td>Sets or returns the binary data string to be used for CAN trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:CAN:FRAMEtype</td>
<td>Sets or returns the CAN trigger frame type</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:CAN:[IDentifier]:ADDRess]:MODE</td>
<td>Sets or returns the CAN addressing mode</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:CAN:[IDentifier]:ADDRess]:VALue</td>
<td>Sets or returns the binary address string used for the CAN trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:CONDITION</td>
<td>Sets or returns the trigger condition for FLEXRAY</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:CYCLEcount:HIValue</td>
<td>Sets or returns the binary data string to be used for FLEXRAY cycle count high value</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:CYCLEcount:QUALifier</td>
<td>Sets or returns the FLEXRAY cycle count qualifier</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:CYCLEcount:VALUE</td>
<td>Sets or returns the binary data string to be used for FLEXRAY cycle count low value</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:DATA:HIVALue</td>
<td>Sets or returns the high binary data string to be used for FLEXRAY trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:DATA:OFFSET</td>
<td>Sets or returns the offset of the data string in bytes</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:DATA:QUALifier</td>
<td>Sets or returns the FLEXRAY data qualifier</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:DATA:SIZE</td>
<td>Sets or returns the length of the data string in bytes</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:DATA:VALUE</td>
<td>Sets or returns the low binary data string to be used for FLEXRAY trigger condition</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:EOFTYPE</td>
<td>Sets or returns the end of file type</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:ERRTYPE</td>
<td>Sets or returns the error type</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:FRAMEID:HIVALue</td>
<td>Sets or returns the binary data string to be used for FLEXRAY frame ID high value</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:FRAMEID:QUALifier</td>
<td>Sets or returns the FLEXRAY frame ID qualifier</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:FRAMEID:VALUE</td>
<td>Sets or returns the binary data string to be used for FLEXRAY frame ID low value</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:FRAMEType</td>
<td>Sets or returns the frame type</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:HEADER:CRC</td>
<td>Sets or returns the CRC portion of the binary header string</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:HEADER:CYCLEcount</td>
<td>Sets or returns the cycle count portion of the binary header string</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:HEADER:FRAMEID</td>
<td>Sets or returns the frame id portion of the binary header string</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:HEADER:INDBits</td>
<td>Sets or returns the indicator bits portion of the binary header string</td>
</tr>
</tbody>
</table>
### Table 2-32: Trigger Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:FLEXray:HEADER:PAYLength</td>
<td>Sets or returns the payload length portion of the binary header string</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:I2C:ADDress:MODE</td>
<td>Sets or returns the I2C address mode to 7 or 10-bit</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:I2C:ADDress:TYPE</td>
<td>Sets or returns the I2C address type to USER</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:I2C:ADDress:VALue</td>
<td>Sets or returns the binary address string used for the I2C trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:I2C:CONDition</td>
<td>Sets or returns the trigger condition for I2C trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:I2C:DATa:DIRection</td>
<td>Sets or returns the I2C trigger condition valid on a READ, WRITE, or either</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:I2C:DATa:SIZE</td>
<td>Sets or returns the length of the data string in bytes to be used for I2C trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:I2C:DATa:VALue</td>
<td>Sets or returns the binary data string used for I2C triggering</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:LIN:CONDition</td>
<td>Sets or returns the trigger condition for LIN</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:LIN:DATa:HIVALue</td>
<td>Sets or returns the binary data string to be used for LIN trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:LIN:DATa:QUALifier</td>
<td>Sets or returns the LIN data qualifier</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:LIN:DATa:SIZE</td>
<td>Sets or returns the length of the data string in bytes to be used for LIN trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:LIN:DATa:VALue</td>
<td>Sets or returns the binary data string</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:LIN:ERTYPE</td>
<td>Sets or returns the error type</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:LIN:IDENTifier:VALue</td>
<td>Sets or returns the binary address string used for LIN trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:PARallel:VALue</td>
<td>Sets or returns the binary data string to be used for a Parallel trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:RS232C:CONDition</td>
<td>Sets or returns the condition for an RS-232C trigger</td>
</tr>
</tbody>
</table>
### Table 2-32: Trigger Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:RS232C:RX:DATa:SIZe</td>
<td>Sets or returns the length of the data string in Bytes for an RX RS-232 Trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:RS232C:RX:DATa:VALue</td>
<td>Sets or returns the binary data string for an RX RS-232 trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:RS232C:TX:DATa:SIZe</td>
<td>Sets or returns the length of the data string for a TX RS-232 trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:RS232C:TX:DATa:VALue</td>
<td>Sets or returns the binary data string for an RS-232 trigger if the trigger condition involves TX</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:SPI:CONDition</td>
<td>Sets or returns the trigger condition for SPI triggering</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:SPI:DATa{:IN</td>
<td>:MISO}:VALue</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:SPI:DATa{:OUT</td>
<td>:MOSI}:VALue</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:SPI:DATa:SIZe</td>
<td>Sets or returns the length of the data string in bytes to be used for SPI trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:ADDRess:HIVALue</td>
<td>Sets or returns the high limit for the USB trigger address</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:ADDRess:VALue</td>
<td>Sets or returns the value for the USB trigger address</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:CONDition</td>
<td>Sets or returns the USB trigger condition</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:DATa:HIVALue</td>
<td>Sets or returns the high limit for the USB trigger data</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:DATa:OFFSet</td>
<td>Sets or returns the data offset for the USB trigger data</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:DATa:SIZe</td>
<td>Sets or returns the number of data bytes for the USB trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:DATa:TYPE</td>
<td>Sets or returns the data type for the USB trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:DATa:VALue</td>
<td>Sets or returns the data value for the USB trigger</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:ENDPoint:VALue</td>
<td>Sets or returns the endpoint value for the USB trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:ERRORTYPE</td>
<td>Sets or returns the error type for the USB trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:HANDSHAKEType</td>
<td>Sets or returns the handshake type for the USB trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:QUALifier</td>
<td>Sets or returns the qualifier for USB trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:SOFFRAMENUMber</td>
<td>Sets or returns the SOF number for the USB trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:SPECIALType</td>
<td>Sets or returns the special packet type for the USB trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:B&lt;x&gt;:USB:TOKENType</td>
<td>Sets or returns the token type for the USB trigger</td>
</tr>
<tr>
<td>TRIGger:A:BUS:SOURce</td>
<td>Sets or returns the source for a bus trigger</td>
</tr>
<tr>
<td>TRIGger:A:EDGE?</td>
<td>Returns the source, coupling and source for the A edge trigger</td>
</tr>
<tr>
<td>TRIGger:A:EDGE:COUPling</td>
<td>Sets or returns the type of coupling for the A edge trigger</td>
</tr>
<tr>
<td>TRIGger:A:EDGE:SLOpe</td>
<td>Sets or returns the slope for the A edge trigger</td>
</tr>
<tr>
<td>TRIGger:A:EDGE:SOURce</td>
<td>Sets or returns the source for the A edge trigger</td>
</tr>
<tr>
<td>TRIGger:A:HOLDoff?</td>
<td>Returns the A trigger holdoff parameters</td>
</tr>
<tr>
<td>TRIGger:A:HOLDoff:TIMe</td>
<td>Sets or returns the A trigger holdoff time</td>
</tr>
<tr>
<td>TRIGger:A:LEVel</td>
<td>Sets or returns the trigger level for the A trigger</td>
</tr>
<tr>
<td>TRIGger:A:LEVel:AUXin</td>
<td>Sets or returns the trigger level for the AUXIN port</td>
</tr>
<tr>
<td>TRIGger:A:LEVel:CH&lt;x&gt;</td>
<td>Specifies or returns the trigger level for the specified trigger channel</td>
</tr>
<tr>
<td>TRIGger:A:LEVel:D&lt;x&gt;</td>
<td>Sets or returns the trigger level for the specified digital channel</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc?</td>
<td>Returns all A trigger logic settings</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc:CLASS</td>
<td>Sets or returns the type of A trigger logic</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc:FUNCTION</td>
<td>Sets or returns the logical combination of the input channels for the A logic trigger</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc:INPUT?</td>
<td>Returns the logic input values for all channels</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc:INPUT:CH&lt;x&gt;</td>
<td>Specifies or returns the logic setting for the specified channel</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc:INPUT:CLK:EDGE</td>
<td>Sets the polarity of the clock channel</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc:INPUT:CLK:SOURce</td>
<td>Sets or returns the channel to use as the clock source</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc:INPUT:D&lt;x&gt;</td>
<td>Sets or returns the logic pattern for a trigger on digital channel &lt;x&gt;</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc:INPUT?</td>
<td>Returns the conditions for generating an A logic pattern trigger</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc:INPUT:DELATime</td>
<td>Sets or returns the pattern trigger delta time value</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc:INPUT:PGM:FINsh:DELTatime</td>
<td>Sets or returns the A logic trigger pattern for a trigger on digital channel &lt;x&gt;</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc:INPUT:WHEN</td>
<td>Sets or returns the pattern logic condition on which to trigger the oscilloscope</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc:INPUT:WHEN:LESSLimit</td>
<td>Sets or returns the maximum time that the selected pattern may be true and still generate an A logic pattern trigger</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc:INPUT:WHEN:MORELimit</td>
<td>Sets or returns the minimum time that the selected pattern may be true and still generate an A logic pattern trigger</td>
</tr>
<tr>
<td>TRIGger:A:LOGIc:THRESHold:CH&lt;x&gt;</td>
<td>Sets or queries the trigger A logic threshold voltage for the specified channel</td>
</tr>
</tbody>
</table>
Table 2-32: Trigger Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger:A:LOGic:THreshold:D&lt;x&gt;</td>
<td>Sets or returns the trigger A logic threshold level for the specified digital channel</td>
</tr>
<tr>
<td>TRIGger:A:LOWerthreshold:CH&lt;x&gt;</td>
<td>Sets or returns the lower threshold for the channel selected</td>
</tr>
<tr>
<td>TRIGger:A:LOWerthreshold:D&lt;x&gt;</td>
<td>Sets the A trigger lower threshold for the digital channel selected</td>
</tr>
<tr>
<td>TRIGger:A:LOWerthreshold{:EXT</td>
<td>:AUX}</td>
</tr>
<tr>
<td>TRIGger:A:MODE</td>
<td>Sets or returns the A trigger mode</td>
</tr>
<tr>
<td>TRIGger:A:PULse?</td>
<td>Returns the A pulse trigger parameters</td>
</tr>
<tr>
<td>TRIGger:A:PULse:CLAss</td>
<td>Sets or returns the type of pulse on which to trigger</td>
</tr>
<tr>
<td>TRIGger:A:PULSEWidth?</td>
<td>Returns the trigger A pulse width parameters</td>
</tr>
<tr>
<td>TRIGger:A:PULSEWidth:POLarity</td>
<td>Sets or returns the polarity for the A pulse width trigger</td>
</tr>
<tr>
<td>TRIGger:A:PULSEWidth:SOUrce</td>
<td>Sets or returns the source for the pulse width trigger</td>
</tr>
<tr>
<td>TRIGger:A:PULSEWidth:WHEn</td>
<td>Sets or returns the criteria for width specification of pulse width trigger events</td>
</tr>
<tr>
<td>TRIGger:A:PULSEWidth:WIDth</td>
<td>Sets or returns the width setting for the pulse width trigger</td>
</tr>
<tr>
<td>TRIGger:A:RUNT?</td>
<td>Returns the current A runt pulse trigger logic parameters</td>
</tr>
<tr>
<td>TRIGger:A:RUNT:POLarity</td>
<td>Sets or returns the polarity for the A pulse runt trigger</td>
</tr>
<tr>
<td>TRIGger:A:RUNT:SOUrce</td>
<td>Sets or returns the source for the A pulse trigger</td>
</tr>
<tr>
<td>TRIGger:A:RUNT:WHEn</td>
<td>Sets or returns the type of pulse width the trigger checks for when it uncovers a runt</td>
</tr>
<tr>
<td>TRIGger:A:RUNT:WIDth</td>
<td>Sets or returns the minimum width for A pulse runt trigger</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>TRIGger:A:SETHold?</td>
<td>Returns settings for setup and hold violation triggering</td>
</tr>
<tr>
<td>TRIGger:A:SETHold:CLOCk?</td>
<td>Returns clock edge polarity, voltage threshold and source input for setup/hold triggering</td>
</tr>
<tr>
<td>TRIGger:A:SETHold:CLOCk:EDGE</td>
<td>Sets or returns the clock edge polarity for setup and hold triggering</td>
</tr>
<tr>
<td>TRIGger:A:SETHold:CLOCk:SOUrce</td>
<td>Sets or returns the clock source for the A logic trigger setup and hold input</td>
</tr>
<tr>
<td>TRIGger:A:SETHold:CLOCk:THReshold</td>
<td>Sets or returns the clock voltage threshold for setup and hold trigger</td>
</tr>
<tr>
<td>TRIGger:A:SETHold:DATa?</td>
<td>Returns the voltage threshold and data source for the setup/hold trigger</td>
</tr>
<tr>
<td>TRIGger:A:SETHold:DATa:SOUrce</td>
<td>Sets or returns the data source for the setup and hold trigger</td>
</tr>
<tr>
<td>TRIGger:A:SETHold:DATa:THReshold</td>
<td>Sets or returns the data voltage threshold for setup and hold trigger</td>
</tr>
<tr>
<td>TRIGger:A:SETHold:HOLDTime</td>
<td>Sets or returns the hold time for the setup and hold violation triggering</td>
</tr>
<tr>
<td>TRIGger:A:SETHold:SETTime</td>
<td>Sets or returns the setup time for setup and hold violation triggering</td>
</tr>
<tr>
<td>TRIGger:A:SETHold:THReshold:CH&lt;x&gt;</td>
<td>Sets or queries the threshold for the channel</td>
</tr>
<tr>
<td>TRIGger:A:SETHold:THReshold:D&lt;x&gt;</td>
<td>Sets the A trigger setup and hold threshold for the selected digital channel</td>
</tr>
<tr>
<td>TRIGger:A:UPPerthreshold:CH&lt;x&gt;</td>
<td>Sets the upper threshold for the channel selected</td>
</tr>
<tr>
<td>TRIGger:A::{TRANsition</td>
<td>RISEFall}?</td>
</tr>
<tr>
<td>TRIGger:A::{TRANsition</td>
<td>RISEFall}:DELTatime</td>
</tr>
</tbody>
</table>
Table 2-32: Trigger Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger:A[:TRANSition]:RISEFall:POLarity</td>
<td>Sets or returns the polarity for the A pulse transition trigger</td>
</tr>
<tr>
<td>TRIGger:A[:TRANSition]:RISEFall:SOURce</td>
<td>Sets or returns the source for transition trigger</td>
</tr>
<tr>
<td>TRIGger:A[:TRANSition]:RISEFall:WHEn</td>
<td>Sets or returns the relationship of delta time to transitioning signal</td>
</tr>
<tr>
<td>TRIGger:A:TYPe</td>
<td>Sets or returns the type of A trigger</td>
</tr>
<tr>
<td>TRIGger:A:VIDeo?</td>
<td>Returns the video parameters for the A trigger</td>
</tr>
<tr>
<td>TRIGger:A:VIDeo:CUSTom[:FORMat]:TYPE</td>
<td>Sets or returns the video trigger format</td>
</tr>
<tr>
<td>TRIGger:A:VIDeo:CUSTom:LINEPeriod</td>
<td>Sets or queries the line period</td>
</tr>
<tr>
<td>TRIGger:A:VIDeo:CUSTom:SCAN</td>
<td>Sets or returns the horizontal line scan rate of the A video trigger</td>
</tr>
<tr>
<td>TRIGger:A:VIDeo:CUSTom:SYNCInterval</td>
<td>Sets or queries the sync interval</td>
</tr>
<tr>
<td>TRIGger:A:VIDeo:HDtv:FORMat</td>
<td>Sets or returns the HDTV video signal format on which to trigger</td>
</tr>
<tr>
<td>TRIGger:A:VIDeo:HOLDoff:FIELD</td>
<td>Sets or returns the video trigger holdoff</td>
</tr>
<tr>
<td>TRIGger:A:VIDeo:LINE</td>
<td>Sets or returns the video line number on which the oscilloscope triggers</td>
</tr>
<tr>
<td>TRIGger:A:VIDeo:POLarity</td>
<td>Sets or returns the polarity of the A video trigger</td>
</tr>
<tr>
<td>TRIGger:A:VIDeo:SOURce</td>
<td>Sets or returns the polarity of the video trigger</td>
</tr>
<tr>
<td>TRIGger:A:VIDeo:STANdard</td>
<td>Sets or returns the video standard</td>
</tr>
<tr>
<td>TRIGger:A:VIDeo{:SYNC}:FIELD</td>
<td>Sets or returns the video field trigger</td>
</tr>
<tr>
<td>TRIGger:B</td>
<td>Sets the B trigger level to 50% or returns the B trigger parameters</td>
</tr>
<tr>
<td>TRIGger:B:BY</td>
<td>Sets or returns B trigger time or event qualifiers</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>TRIGger:B:EDGE?</td>
<td>Returns B trigger edge type parameters</td>
</tr>
<tr>
<td>TRIGger:B:EDGE:COUpllng</td>
<td>Sets or returns the type of B trigger coupling</td>
</tr>
<tr>
<td>TRIGger:B:EDGE:SLOpe</td>
<td>Sets or returns the B edge trigger slope</td>
</tr>
<tr>
<td>TRIGger:B:EDGE:SOUrce</td>
<td>Sets or returns the B edge trigger source</td>
</tr>
<tr>
<td>TRIGger:B:EVENTS?</td>
<td>Returns the current B trigger events parameter</td>
</tr>
<tr>
<td>TRIGger:B:EVENTS:COUNt</td>
<td>Sets or returns the number of events that must occur before the B trigger occurs</td>
</tr>
<tr>
<td>TRIGger:B:LEVel</td>
<td>Sets or returns the level for the B trigger</td>
</tr>
<tr>
<td>TRIGger:B:LEVel:CH&lt;x&gt;</td>
<td>Sets or returns the level for the B trigger for a specific channel</td>
</tr>
<tr>
<td>TRIGger:B:LEVel:D&lt;x&gt;</td>
<td>Sets or returns the B trigger level for digital channel &lt;x&gt;</td>
</tr>
<tr>
<td>TRIGger:B:LOWerthreshold:CH&lt;x&gt;</td>
<td>Sets or returns the B trigger lower threshold for the channel selected</td>
</tr>
<tr>
<td>TRIGger:B:LOWerthreshold:D&lt;x&gt;</td>
<td>Sets or queries the B trigger lower threshold for the digital channel selected</td>
</tr>
<tr>
<td>TRIGger:B:STATE</td>
<td>Returns the current state of the B trigger</td>
</tr>
<tr>
<td>TRIGger:B:TIME</td>
<td>Sets or returns the B trigger delay time</td>
</tr>
<tr>
<td>TRIGger:B:TYPe</td>
<td>Sets or returns the type of B trigger</td>
</tr>
<tr>
<td>TRIGger:B:UPPerthreshold:CH&lt;x&gt;</td>
<td>Sets or returns the B trigger upper threshold for the channel selected</td>
</tr>
<tr>
<td>TRIGger:EXTernal?</td>
<td>Returns external trigger parameters</td>
</tr>
<tr>
<td>TRIGger:EXTernal:PRObe</td>
<td>Sets or returns the attenuation factor value of the external probe connector</td>
</tr>
<tr>
<td>TRIGger:EXTernal:YUNIts?</td>
<td>Returns the external trigger vertical (Y) units value</td>
</tr>
</tbody>
</table>
Table 2-32: Trigger Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger:FREQuency?</td>
<td>Returns the trigger frequency in hertz, if available</td>
</tr>
<tr>
<td>TRIGger:STATE?</td>
<td>Returns the current state of the triggering system</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 display of channel, reference, and math waveforms.

Table 2-33: Vertical Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUXin?</td>
<td>Returns auxiliary input parameters</td>
</tr>
<tr>
<td>AUXin:PRObe</td>
<td>Returns all information concerning the probe attached to auxiliary input</td>
</tr>
<tr>
<td>AUXin:PRObe:AUTOZero</td>
<td>Sets the TekVPI probe attached to the auxiliary input to autozero</td>
</tr>
<tr>
<td>AUXin:PRObe:COMMAND</td>
<td>Sets the state of the specified probe control</td>
</tr>
<tr>
<td>AUXin:PRObe:DEGAUss</td>
<td>Starts a degauss/autozero cycle on a TekVPI current probe attached to the auxiliary input</td>
</tr>
<tr>
<td>AUXin:PRObe:DEGAUss:STATE?</td>
<td>Returns the degauss state of the TekVPI current probe attached to the auxiliary input</td>
</tr>
<tr>
<td>AUXin:PRObe:FORCEDRange</td>
<td>Sets or returns the range of the TekVPI probe attached to the auxiliary input</td>
</tr>
<tr>
<td>AUXin:PRObe:GAIN</td>
<td>Sets or returns the gain factor of the probe that is attached to the auxiliary input</td>
</tr>
<tr>
<td>AUXin:PRObe:ID:SERnumber?</td>
<td>Returns the serial number of the probe that is attached to the auxiliary input</td>
</tr>
<tr>
<td>AUXin:PRObe:ID:TYPE?</td>
<td>Returns the type of probe that is attached to the auxiliary input</td>
</tr>
<tr>
<td>AUXin:PRObe:RESistance?</td>
<td>Returns the resistance of the probe that is attached to the Auxiliary input</td>
</tr>
<tr>
<td>AUXin:PRObe:SIGnal</td>
<td>Sets or returns the input bypass setting on VPI probes that support input bypass</td>
</tr>
<tr>
<td>AUXin:PRObe:UNIts?</td>
<td>Returns the units of measure of the probe that is attached to the auxiliary input</td>
</tr>
<tr>
<td>CH&lt;xx&gt;?</td>
<td>Returns vertical parameters for the specified channel</td>
</tr>
</tbody>
</table>
### Table 2-33: Vertical Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH&lt;x&gt;:AMPSVIAVOLTs:ENAble</td>
<td>Sets or returns the state of the amps via volts feature for the specified channel</td>
</tr>
<tr>
<td>CH&lt;x&gt;:AMPSVIAVOLTs:FACTOR</td>
<td>Sets or returns the amps via volts factor for the specified channel</td>
</tr>
<tr>
<td>CH&lt;x&gt;:BANDwidth</td>
<td>Sets or returns the bandwidth of the specified channel</td>
</tr>
<tr>
<td>CH&lt;x&gt;:COUPling</td>
<td>Sets or returns the coupling setting for the specified channel</td>
</tr>
<tr>
<td>CH&lt;x&gt;:DESKew</td>
<td>Sets or returns the deskew time for the specified channel</td>
</tr>
<tr>
<td>CH&lt;x&gt;:INVert</td>
<td>Sets or returns the invert function for the specified channel</td>
</tr>
<tr>
<td>CH&lt;x&gt;:LABel</td>
<td>Sets or returns the waveform label for channel &lt;x&gt;</td>
</tr>
<tr>
<td>CH&lt;x&gt;:OFFSet</td>
<td>Sets or returns the channel offset</td>
</tr>
<tr>
<td>CH&lt;x&gt;:POSition</td>
<td>Sets or returns the channel vertical position</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe?</td>
<td>Returns the gain, resistance, units, and ID of the probe that is attached to the specified channel</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:AUTOZero</td>
<td>Sets the TekVPI probe attached to the specified channel input to autozero</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:COMMAND</td>
<td>Sets the state of the specified probe control</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:DEGAUss</td>
<td>Starts a degauss/autozero cycle on a TekVPI current probe attached to the specified channel input</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:DEGAUss:STATE?</td>
<td>Returns the state of the probe degauss</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:FORCEDRange</td>
<td>Sets or returns the range on a TekVPI probe attached to the specified channel</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:GAIN</td>
<td>Sets or returns the gain factor 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>
<tr>
<td>CH&lt;x&gt;:PRObe:MODel</td>
<td>Sets or returns the probe model for the specified channel</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:PROPDELay</td>
<td>Sets or returns the propagation delay for the probe connected to the specified channel</td>
</tr>
</tbody>
</table>
### Table 2-33: Vertical Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH&lt;x&gt;:PRObe:RECDESkew?</td>
<td>Returns the recommended deskew for the probe connected to the specified channel</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:RESistance?</td>
<td>Returns the resistance of the probe that is attached to the specified channel</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:SIGnal</td>
<td>Sets or returns the input bypass setting of channel &lt;x&gt; TekVPI probe</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PRObe:UNIts?</td>
<td>Returns the units of measure of the probe that is attached to the specified channel</td>
</tr>
<tr>
<td>CH&lt;x&gt;:SCAle</td>
<td>Sets or returns the vertical scale of the specified channel</td>
</tr>
<tr>
<td>CH&lt;x&gt;:TERmination</td>
<td>Sets or returns channel input termination</td>
</tr>
<tr>
<td>CH&lt;x&gt;:YUNits</td>
<td>Sets or returns the units for the specified channel</td>
</tr>
<tr>
<td>D&lt;x&gt;</td>
<td>Returns parameters for digital channel &lt;x&gt;</td>
</tr>
<tr>
<td>D&lt;x&gt;:LABel</td>
<td>Sets or returns the waveform label for digital channel &lt;x&gt;</td>
</tr>
<tr>
<td>D&lt;x&gt;:THREshold</td>
<td>Sets or returns the logical threshold for digital channel &lt;x&gt;</td>
</tr>
<tr>
<td>D&lt;x&gt;:POSition</td>
<td>Sets or returns the vertical position for digital channel &lt;x&gt;</td>
</tr>
<tr>
<td>DESkew</td>
<td>Causes the deskew values for all channels to be set to the recommended values</td>
</tr>
<tr>
<td>DESkew:DISPlay</td>
<td>Sets or returns the state of the deskew table display</td>
</tr>
<tr>
<td>REF&lt;x&gt;?</td>
<td>Returns reference waveform data for channel &lt;x&gt;</td>
</tr>
<tr>
<td>REF&lt;x&gt;:DATE?</td>
<td>Returns the date that a reference waveform was stored</td>
</tr>
<tr>
<td>REF&lt;x&gt;:HORizont:DELay:TIMe</td>
<td>Sets or returns the horizontal position of the specified reference waveform in percent of the waveform that is displayed to the right of the center vertical graticule</td>
</tr>
<tr>
<td>REF&lt;x&gt;:HORizont:SCAle</td>
<td>Sets or returns the horizontal scale for a reference waveform</td>
</tr>
<tr>
<td>REF&lt;x&gt;:LABel</td>
<td>Sets or returns the specified reference waveform label</td>
</tr>
<tr>
<td>REF&lt;x&gt;:VERTical:POSition</td>
<td>Sets or returns the vertical position for reference channel &lt;x&gt;</td>
</tr>
<tr>
<td>REF&lt;x&gt;:TI Ме?</td>
<td>Returns the time that a reference waveform was stored</td>
</tr>
<tr>
<td>REF&lt;x&gt;:VERTical:POSition</td>
<td>Sets or returns the vertical position of the specified reference waveform</td>
</tr>
</tbody>
</table>
Table 2-33: Vertical Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REF&lt;x&gt;:VERTical:SCAle</td>
<td>Sets or returns the reference waveform vertical scale in vertical units/div</td>
</tr>
<tr>
<td>SELECT</td>
<td>Returns information on which waveforms are on or off and which waveform is selected.</td>
</tr>
<tr>
<td>SELECT:BUS&lt;x&gt;</td>
<td>Turns on or off the specified bus waveform or returns whether the specified bus channel is on or off</td>
</tr>
<tr>
<td>SELECT:CH&lt;x&gt;</td>
<td>Turns on or off the specified waveform or returns whether the specified channel is on or off</td>
</tr>
<tr>
<td>SELECT:CONTROL</td>
<td>Sets or returns the waveform that is selected as the implied recipient of channel-related commands</td>
</tr>
<tr>
<td>SELECT:D&lt;x&gt;</td>
<td>Turns on the display of digital channel &lt;x&gt; and resets the acquisition</td>
</tr>
<tr>
<td>SELECT:MATH[1]</td>
<td>Turns on or off the math waveform or returns whether the math waveform is on or off</td>
</tr>
<tr>
<td>SELECT:REF&lt;x&gt;</td>
<td>Turns on or off the specified reference waveform or returns whether the specified reference waveform is on or off</td>
</tr>
</tbody>
</table>

Waveform Transfer Command Group

Use the commands in the Waveform Transfer Command Group to transfer waveform data points to and from the oscilloscope. 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 or destination.

Data Formats

All data points for DPO and MSO models are signed integer format only. Valid data widths for CH1-CH4, MATH, REF1-REF4 and D0-D15 are 1 and 2-byte widths. The valid data widths for the digital collection is either 4 or 8-byte widths.

The oscilloscope can transfer waveform data in either ASCII or binary format. You specify the format with the DATa:ENCdg command.

ASCII Data. ASCII data is represented by signed integer values. The range of the values depends on the byte width specified. One byte wide data ranges from -128 to 127. Two byte wide data ranges from -32768 to 32767.
Each data value requires two to seven characters. This includes one to five characters to represent the value, another character, if the value is negative, to represent a minus sign, and a comma to separate the data points.

An example ASCII waveform data string may look like this:

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

**NOTE.** You can use ASCII to obtain a readable and easier to format output than binary. However, the oscilloscope may require bytes to send the same values with ASCII than with binary, reducing transmission speed.

The use of ASCII for waveform data transfer is inefficient. ASCII-formatted Waveform (WAVFRM?) and Curve (CURVE?) queries, exceeding 1 M points, are not supported.

### Binary Data

Binary data is represented by signed integer or positive integer 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, and positive integer values range from 0 to 65,535.

<table>
<thead>
<tr>
<th>Byte width</th>
<th>Signed integer range</th>
<th>Positive integer range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-128 to 127</td>
<td>0 to 255</td>
</tr>
<tr>
<td>2</td>
<td>32,768 to 32,767</td>
<td>0 to 65,535</td>
</tr>
</tbody>
</table>

The defined binary formats also specify the order in which the bytes are transferred. The four binary formats are RIBinary, RPBinary, SRIBinary, and SRPBinary.

RIBinary is signed integer where the most significant byte is transferred first, and RPBinary is positive integer where the most significant byte is transferred first. SRIBinary and SRPBinary correspond to RIBinary and RPBinary respectively but use a swapped byte order where the least significant byte is transferred first. The byte order is ignored when DATa:WIDth is set to 1.

### 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 into the oscilloscope, you must first specify the record length of the destination waveform record. You do this with the WFMInpre:NR_Pt
command. Next, specify the first data point within the waveform record. For example, when you set DATa:STARt to 1, data points will be stored starting with the first point in the record. The oscilloscope will ignore the value set by DATa:STOP when reading in data. It will stop reading data when there is no data to read or when it has reached the specified record length.

When transferring data from the oscilloscope, 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 oscilloscope. You can only transfer one waveform at a time. Waveforms sent to the oscilloscope are always stored in one of the reference memory locations. Use the DATa:DESTination command to specify a reference memory location.

Waveform Preamble

Each waveform you transfer has an associated waveform preamble, which contains information such as horizontal scale, vertical scale, and the other settings in effect when the waveform was created. Refer to the individual WFMInpre and WFMOutpre commands for 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 Oscilloscope

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

1. Select the waveform source(s) using DATa:SOUrce.
2. Specify the waveform data format using DATa:ENCdg.
3. Specify the number of bytes per data point using WFMOutpre:BYT_Nr.
4. Specify the portion of the waveform that you want to transfer using DATa:STARt and DATa:STOP.
5. Transfer waveform preamble information using the WFMOutpre? query.
6. Transfer waveform data from the oscilloscope using the CURVe? query.
Transferring Waveform Data to the Oscilloscope

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

1. Specify the reference waveform using DATa:DESTination.
2. Specify the record length of the reference waveform using WFMPre:NR_Pt.
3. Specify the waveform data format using WFMInpre:ENCdg.
4. Specify the number of bytes per data point using WFMInpre:BYT_Nr.
5. Specify first data point in the waveform record using DATa:STARt.
6. Transfer waveform preamble information using WFMInpre.
7. Transfer waveform data to the oscilloscope using CURVe.

Table 2-35: Waveform Transfer Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURVe</td>
<td>The command format transfers waveform data to the oscilloscope (reference waveform specified by DATa:DESTination) The query format transfers waveform data from oscilloscope specified by the DATa:SOUrce command</td>
</tr>
<tr>
<td>DATa</td>
<td>Sets the format and location of the waveform data that is transferred with the CURVe Command Or returns the format and location of the waveform data that is transferred with the CURVe? command</td>
</tr>
<tr>
<td>DATa:DESTination</td>
<td>Sets or returns the reference waveform for storing waveform data sent to the oscilloscope</td>
</tr>
<tr>
<td>DATa:ENCdg</td>
<td>Sets or returns the format of outgoing waveform data</td>
</tr>
<tr>
<td>DATa:SOUrce</td>
<td>Sets or returns the location of waveform data transferred from the oscilloscope</td>
</tr>
<tr>
<td>DATa:STARt</td>
<td>Sets or returns the starting point in waveform transfer</td>
</tr>
<tr>
<td>DATa:STOP</td>
<td>Sets or returns the ending data point in waveform transfer</td>
</tr>
<tr>
<td>WAVFrm?</td>
<td>Returns a branch query containing waveform data in either binary or ASCII format, waveform formatting data, and the location of the waveform data source</td>
</tr>
<tr>
<td>WFMInpre?</td>
<td>Returns the waveform formatting specification to be applied to the next incoming CURVE command data</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WFMInpre:BIT_Nr</td>
<td>Sets or returns the number of bits per binary waveform point for the incoming waveform</td>
</tr>
<tr>
<td>WFMInpre:BN_Fmt</td>
<td>Sets or returns the format of binary data for the incoming waveform</td>
</tr>
<tr>
<td>WFMInpre:BYT_Nr</td>
<td>Sets or returns the data width for the incoming waveform</td>
</tr>
<tr>
<td>WFMInpre:BYT_Or</td>
<td>Sets or returns the byte order of waveform points for the incoming waveform</td>
</tr>
<tr>
<td>WFMInpre:ENCdg</td>
<td>Sets or returns the type of encoding for incoming waveform data</td>
</tr>
<tr>
<td>WFMInpre:NR_Pt</td>
<td>Sets or returns the number of points in the incoming waveform record</td>
</tr>
<tr>
<td>WFMInpre:PT_Fmt</td>
<td>Sets or returns the point format of incoming waveform data</td>
</tr>
<tr>
<td>WFMInpre:PT_Off</td>
<td>This query always returns a 0</td>
</tr>
<tr>
<td>WFMInpre:XINcr</td>
<td>Sets or returns the horizontal sampling interval between incoming waveform points</td>
</tr>
<tr>
<td>WFMInpre:XUInit</td>
<td>Sets or returns the horizontal units of the incoming waveform</td>
</tr>
<tr>
<td>WFMInpre:XZEro</td>
<td>Sets or returns the time of the first point in the incoming waveform</td>
</tr>
<tr>
<td>WFMInpre:YMUlt</td>
<td>Sets or returns the vertical scale factor, per digitizing level, of the incoming waveform points</td>
</tr>
<tr>
<td>WFMInpre:YOff</td>
<td>Sets or returns the vertical position of the incoming waveform in digitizing levels</td>
</tr>
<tr>
<td>WFMInpre:YUInit</td>
<td>Sets or returns the vertical units of the incoming waveform</td>
</tr>
<tr>
<td>WFMInpre:YZEro</td>
<td>Sets or returns the vertical offset of the incoming waveform</td>
</tr>
<tr>
<td>WFMOutpre?</td>
<td>Returns the waveform formatting data for the waveform specified by the DATA:SOURCE command</td>
</tr>
<tr>
<td>WFMOutpre:BIT_Nr</td>
<td>Sets or returns the number of bits per waveform point that outgoing waveforms contain</td>
</tr>
<tr>
<td>WFMOutpre:BN_Fmt</td>
<td>Sets or returns the format of binary data for the outgoing waveform</td>
</tr>
<tr>
<td>WFMOutpre:BYT_Nr</td>
<td>Sets or returns the data width for the outgoing waveform</td>
</tr>
<tr>
<td>WFMOutpre:BYT_Or</td>
<td>Sets or returns the byte order of waveform points for the outgoing waveform</td>
</tr>
</tbody>
</table>
Table 2-35: Waveform Transfer Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFMOutpre:ENCdg</td>
<td>Sets or returns the type of encoding for outgoing waveforms</td>
</tr>
<tr>
<td>WFMOutpre:FRACTional?</td>
<td>This query always returns a 0 if the waveform specified by DATA:SOURce is on or displayed</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 outgoing waveform</td>
</tr>
<tr>
<td>WFMOutpre:PT_OFF?</td>
<td>This query always returns a 0 if the waveform specified by DATA:SOURce is on or displayed</td>
</tr>
<tr>
<td>WFMOutpre:PT_ORDER?</td>
<td>This query always returns LINEAR</td>
</tr>
<tr>
<td>WFMOutpre:WFId?</td>
<td>Returns a string describing the acquisition parameters for the outgoing waveform</td>
</tr>
<tr>
<td>WFMOutpre:XIINcr?</td>
<td>Returns the horizontal sampling interval for the outgoing waveform</td>
</tr>
<tr>
<td>WFMOutpre:XUNIT?</td>
<td>Returns the horizontal units for the outgoing waveform</td>
</tr>
<tr>
<td>WFMOutpre:XZERO?</td>
<td>Returns the time of the first point in the outgoing waveform</td>
</tr>
<tr>
<td>WFMOutpre:YMULT?</td>
<td>Returns the vertical scale factor per digitizing level for the outgoing waveform</td>
</tr>
<tr>
<td>WFMOutpre:YOFF?</td>
<td>Returns the vertical position in digitizing levels for the outgoing waveform</td>
</tr>
<tr>
<td>WFMOutpre:YUNIT?</td>
<td>Returns the vertical units for the outgoing waveform</td>
</tr>
<tr>
<td>WFMOutpre:YZERO?</td>
<td>Returns the vertical offset for the outgoing waveform</td>
</tr>
</tbody>
</table>

Zoom Command Group

Use the commands in the Zoom Command Group to expand and position the waveform display horizontally, without changing the time base settings.

Table 2-36: Zoom Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZOOM?</td>
<td>Returns the horizontal positioning and scaling of the zoom display</td>
</tr>
<tr>
<td>ZOOM:MODE</td>
<td>Sets or returns the zoom mode</td>
</tr>
<tr>
<td>ZOOM:ZOOM&lt;x&gt;?</td>
<td>Returns the current horizontal positioning and scaling of the zoom display</td>
</tr>
</tbody>
</table>
### Table 2-36: Zoom Commands (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZOOM:ZOOM&lt;x&gt;:FACTOR?</td>
<td>Returns the zoom factor of the zoom window. &lt;x&gt; can only be 1</td>
</tr>
<tr>
<td>ZOOM:ZOOM&lt;x&gt;:POSITION</td>
<td>Sets or returns the horizontal position of the zoom window in terms of 0 to 100% of the overview window. &lt;x&gt; can only be 1</td>
</tr>
<tr>
<td>ZOOM:ZOOM&lt;x&gt;:SCALE</td>
<td>Sets or returns the horizontal zoom scale of the zoom window. &lt;x&gt; can only be 1</td>
</tr>
<tr>
<td>ZOOM:ZOOM&lt;x&gt;:STATE</td>
<td>Specifies or returns a trace as zoomed, on or off. &lt;x&gt; can only be 1</td>
</tr>
</tbody>
</table>
Command Groups
Commands Listed in Alphabetical Order

ACQuire? (Query Only)

Returns the following current acquisition parameters:

- Stop after
- Acquisition state
- Mode
- Number of averages
- Sampling mode

Group  Acquisition
Syntax    ACQuire?

Related Commands  ACQuire:MODe, ACQuire:NUMACq?, ACQuire:NUMAVg, ACQuire:STOPAfter

ACQuire:MAgnivu

Sets or returns the MagniVu feature, which provides up to 32 times signal detail for fast viewing of short events. This feature is not recommended for slow data formats such as RS-232.

**NOTE.** MagniVu channel sampling is available on MSO oscilloscopes only.

Group  Acquisition
Syntax  ACQuire:MAgnivu {<NR1>|OFF|ON}
        ACQuire:MAgnivu?
Arguments  <NR1> = 0 disables the MagniVu feature; any other value turns this feature on.
            ON enables the MagniVu feature.
            OFF disables the MagniVu feature.
\textbf{ACQuire:MAXSamplerate? (Query Only)}

Returns the maximum real-time sample rate, which varies from model to model.

\begin{itemize}
  \item **Group**: Acquisition
  \item **Syntax**: ACQuire:MAXSamplerate?
  \item **Examples**: ACQUIRE:MAXSAMPLERATE? might return 2.5000E+9 in a DPO4034 indicating the maximum real-time sample rate is 2.5GS/s.
\end{itemize}

\textbf{ACQuire:MODE}

Sets or returns the acquisition mode of the oscilloscope for all live waveforms.

Waveforms are the displayed data point values taken from acquisition intervals. Each acquisition interval represents a time duration set by the horizontal scale (time per division). The oscilloscope sampling system always samples at the maximum rate, so the acquisition interval may include than one sample.

The acquisition mode (which you set using this ACQuire:MODE command) determines how the final value of the acquisition interval is generated from the many data samples.

\begin{itemize}
  \item **Group**: Acquisition
  \item **Syntax**: ACQuire:MODE \{SAMple|PEAKdetect|HIRes|AVErage|ENvelope\}
  \item **ACQuire:MODE?**
  \item **Related Commands**: ACQuire:NUMAVg, CURVe
  \item **Arguments**
    \begin{itemize}
      \item \texttt{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.
      \item \texttt{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.
      \item \texttt{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 oscilloscope 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 peak data points from every waveform acquisition.

### 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)

Returns the number of waveform acquisitions that have occurred since starting acquisition with the **ACQuire:STATE:RUN** command. This value is reset to zero when any acquisition, horizontal, or vertical arguments that affect the waveform are changed. The maximum number of acquisitions that can be counted is $2^{32} - 1$.

#### Group
Acquisition

#### Syntax

**ACQuire:NUMACq**

#### Related Commands

**ACQuire:STATE**

#### Returns

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

### ACQuire:NUMAVg

Sets or returns the number of waveform acquisitions that make up an averaged waveform. Use the **ACQuire:MODe** command to enable the Average mode. Sending this command is equivalent to turning a multipurpose knob to enter the number of waveform acquisitions to average.
Commands Listed in Alphabetical Order

**Group**  
Acquisition

**Syntax**  
ACQuire:NUMAVg <NR1>  
ACQuire:NUMAVg?

**Related Commands**  
ACQuire:MODE

**Arguments**  
<NR1> is the number of waveform acquisitions to average. The range of values is from 2 to 512 in powers of two.

**Examples**  
ACQUIRE:NUMAVG 16 specifies that 16 waveform averages will be performed before exponential averaging starts.

ACQUIRE:NUMAVG? might return :ACQUIRE:NUMAVG 64 indicating that there are 64 acquisitions specified for averaging.

**ACQuire:STATE**

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 oscilloscope resets the number of acquisitions. If the RUN argument is issued while in continuous mode, acquisition continues.

**Group**  
Acquisition

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

ACQuire:STATE?

**Related Commands**  
ACQuire:STOPAfter

**Arguments**  
OFF stops acquisitions.

STOP stops acquisitions.

ON starts acquisitions.

RUN starts acquisitions.
<NR1> = 0 stops acquisitions; any other value starts 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:STOPAAfter**

Sets or returns whether the oscilloscope continually acquires acquisitions or acquires a single sequence.

**Group**

Acquisition

**Syntax**

ACQuire:STOPAAfter {RUNSTOP|SEQUence}

ACQuire:STOPAAfter?

**Related Commands**

ACQuire:STATE

**Arguments**

RUNSTOP specifies that the oscilloscope 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 oscilloscope to continually acquire data.

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

**ALIas**

Sets or returns the state of alias functionality. Use Alias commands to define new commands as a sequence of standard commands. You may 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.

To use Alias commands, first define the alias, then turn on the alias state.

**Group**

Alias
Syntax

```
ALias {OFF|ON|<NR1>}
ALias?
```

Related Commands

- ALias:DEFine
- ALias[:STATE]

Arguments

- **OFF** turns alias expansion off. If a defined alias is sent when ALias is off, a command error (110) will be generated.
- **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 1 indicating that the alias feature is on.

**ALias:CATalog? (Query Only)**

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**

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 has been 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.
Commands Listed in Alphabetical Order

Group

Alias

Syntax

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

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 5;:AUTOSET
EXECUTE;:SELECT:CH1 ON" defines an alias named "ST1" that sets
up the oscilloscope.

ALIAS:DEFINE? "ST1" returns ALIAS:DEFINE "ST1",#246
:RECALL:SETUP 5;:AUTOSET EXECUTE;:SELECT:CH1 ON

ALIas:DELEte (No Query Form)

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)**

Deletes all existing aliases.

Group: Alias

Syntax: ALIAS:DELETE:ALL

Related Commands: ALIAS:DELETE, ALIAS:DELETE[:NAME]

Examples

ALIAS:DELETE:ALL deletes all existing aliases.

**ALIAS:DELETE[:NAME] (No Query Form)**

Removes a specified alias. This command is identical to ALIAS:DELETE.

Group: Alias

Syntax: ALIAS:DELETE[:NAME] <QString>

Arguments: <QString> is the name of the alias to remove. Using ALIAS:DELETE[:NAME] without specifying an alias causes an execution error. <QString> must be an existing alias.

Examples

ALIAS:DELETE[:NAME] “STARTUP” deletes the alias named STARTUP.

**ALIAS[:STATE]**

Turns aliases on or off. This command is identical to the ALIAS command.

Group: Alias

Syntax: ALIAS[:STATE] {<NR1>|OFF|ON}
ALIAS[:STATE]?
Arguments

OFF or \(<\text{NR1}> = 0\) turns alias expansion off. If a defined alias is sent when \text{ALIAS}:\text{STATE} is OFF, a command error (102) is generated.

ON or \(<\text{NR1}> 0\) turns alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

Examples

\text{ALIAS}[>:\text{STATE}] \text{OFF} turns the command alias feature off.

\text{ALIAS}[>:\text{STATE}]? returns 0 when the alias feature is off.

\textbf{ALLEV? (Query Only)}

Prompts the oscilloscope to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue. Use the \text{*ESR?} query to enable the events to be returned. This command is similar to repeatedly sending \text{*EVMsg?} queries to the oscilloscope.

Group

Status and Error

Syntax

\text{ALLEV?}

Related Commands

\text{*ESR?, EVMsg?}

Examples

\text{ALLEV?} might return \text{:ALLEV 2225,}"\text{Measurement error, No waveform to measure;} \"420,\text{Query UTERMINATED;}'

\textbf{AUTOSet (No Query Form)}

Sets the vertical, horizontal, and trigger controls of the oscilloscope to automatically acquire and display the selected waveform.

Group

Miscellaneous

Syntax

\text{AUTOSet \{EXECute|UNDo\}}

Arguments

\text{EXECute} autosets the displayed waveform.

\text{UNDo} restores the oscilloscope settings to those present prior to the autoset execution.
Examples
AUTOSET EXECUTE vertical, horizontal, and trigger controls of the oscilloscope to automatically acquire and display the selected waveform.

**AUTOSet:ENAble**

Enables or disables the autoset 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. This setting is not saved in setup files or SET? or *LRN? queries. The default state is 1 (autoset enabled).

**Group**
Miscellaneous

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

**Arguments**
OFF or 0 disables autoset.
ON or 1 enables autoset.

**Examples**
AUTOSET:ENABLE 1 enables the front-panel Autoset button.
AUTOSET:ENABLE? might return AUTOSET:ENABLE 0 indicating that the autoset feature is disabled.

**AUXin?** *(Query Only)*

Returns all auxiliary input parameters.

**Group**
Vertical

**Syntax**
AUXin?

**AUXin:PRObe**

Returns all information concerning the probe attached to auxiliary input.

**Group**
Vertical

**Syntax**
AUXin:PRObe
AUXin:PRObe?

**AUXin:PRObe:AUTOZero (No Query Form)**

Sets the TekVPI probe attached to the Aux In input to autozero. The oscilloscope will ignore this command if the Auxiliary input does not have a TekVPI probe connected to it.

**Group** Vertical

**Syntax** AUXin:PRObe:AUTOZero {EXECute}

**Arguments** EXECute sets the probe to autozero.

**Examples** AUXin:PROBE:AUTOZERO EXECUTE

**AUXin:PRObe:COMMAND (No Query Form)**

Sets the state of the probe control specified with the first argument to the state specified with the second argument. The commands and states are unique to the attached probe type. Only certain VPI probes support this command. See the probe documentation for how to set these string arguments.

**Group** Vertical

**Syntax** AUXin:PRObe:COMMAND <QString>, <QString>

**Arguments** <QString> are quoted strings specifying the probe command and value to set in the probe attached to the auxiliary input.

**Examples** AUXIN:PROBE:COMMAND “OUTPUT”, “ON” turns the output of a Tektronix VPI-DPG probe on.

AUXIN:PROBE:COMMAND “MODE”, “4–4V1MHz” sets a Tektronix VPI-DPG probe to the 4-4V1MHz mode.

AUXIN:PROBE:COMMAND?”MODE” might return AUXIN:PROBE:COMMAND “MODE”, “4–4V1MHz”.

MSO4000 and DPO4000 Series Programmer Manual 2-81
**AUXin:PRObe:DEGAUss (No Query Form)**

Starts a degauss/autozero cycle on a TekVPI current probe attached to the Aux In input. If you send this command to a probe that does not support this function, it is ignored.

**Syntax**

AUXin:PRObe:DEGAUss {EXECute}

**Arguments**

EXECute starts a probe degauss cycle.

**Examples**

AUXin:PROBE:DEGAUSS EXECUTE degausses the probe attached to the Aux In input.

---

**AUXin:PRObe:DEGAUss:STATE? (Query Only)**

Returns the state of the probe degauss (NEEDED, RECOMMENDED, PASSED, FAILED, RUNNING). The command will return PASSED for probes that do not support degauss operations.

**Syntax**

AUXin:PRObe:DEGAUss:STATE?

**Examples**

AUXin:PROBE:DEGAUSS:STATE? might return:

AUXin:PROBE:DEGAUSS:STATE PASSED indicating that the probe has been degaussed.

---

**AUXin:PObe:FORCEDRange**

Changes or returns the range on a TekVPI probe attached to the Aux In input.

**Syntax**

AUXin:PRObe:FORCEDRange <NR3>

AUXin:PRObe:FORCEDRange?
Arguments  

 `<NR3>` is the probe range, which is probe dependent.

**AUXin:PRObe:GAIN**

Sets or returns the gain factor of a probe that is attached to the Aux In input.

**Group**  

Vertical

**Syntax**  

AUXin:PRObe:GAIN `<NR3>`  
AUXin:PRObe:GAIN?

**Arguments**  

`<NR3>` is the probe gain, which is probe dependent.

**Examples**  

indicating that the attached 10x probe delivers 0.1 V to the Aux In BNC for every 1.0 V applied to the probe input.

**AUXin:PRObe:ID:SERnumber? (Query Only)**

Returns the serial number of the probe that is attached to the auxiliary input.

**Group**  

Vertical

**Syntax**  

AUXin:PRObe:ID:SERnumber?

**AUXin:PRObe:ID:TYPE? (Query Only)**

Returns the type of probe that is attached to the auxiliary input.

**Group**  

Vertical

**Syntax**  

AUXin:PRObe:ID:TYPE?

**AUXin:PRObe:RESistance? (Query Only)**

Returns the resistance of the probe attached to the front panel Aux In connector.
AUXin:PRObe:RESistance?

**Examples**

AUXin:PRObe:RESistance? might return :AUXin:PROBE:RESISTANCE 1.0000E+6 indicating that the input resistance of the probe attached to the front panel Aux In connector is 1 MΩ.

**NOTE.** This query will return 0.0 if no probe is attached or the attached probe does not report the input resistance.

AUXin:PRObe:SIGnal

This command changes the input bypass setting on VPI probes that support input bypass, for example the TCP0001. If sent to a probe that does not support input bypass, it is ignored.

**Arguments**

- **ByPass** sets the probe to Bypass mode.
- **PASS** sets the probe to Pass mode.

AUXin:PRObe:UNIts? (Query Only)

Returns a string describing the units of measure of the probe attached to the Aux In input.

**Examples**

AUXin:PROBE:UNITS? might return: :AUXin:PROBE:UNITS “V” indicating that the units of measure for the attached probe are volts.
**BUS? (Query Only)**

Returns the parameters for each bus. These parameters affect either the Serial Trigger Setup or the Bus Display.

**Conditions**
This command requires a serial bus analysis module.

**Group**
Bus

**Syntax**
BUS?

---

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

Sets or returns the number of delay bits for the AUDIO bus.

**NOTE.** This command is applicable only for TDM audio bus type (BUS:B<x>AUDio:TYPE TDM)

**Conditions**
This command requires DPO4AUDIO application module.

**Group**
Bus

**Syntax**

- BUS:B<x>:AUDio:BITDelay <NR1>
- BUS:B<x>:AUDio:BITDelay?

**Arguments**
<NR1> specifies the number of delay bits.

**Examples**

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

---

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

Sets or returns the bit order for the AUDIO bus.

**Conditions**
This command requires DPO4AUDIO application module.
Commands Listed in Alphabetical Order

**Group**  Bus

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

**Arguments**  
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:B<x>:AUDio:CHANnel:SIZE**  
Sets or returns the number of bits per channel for the AUDIO bus.  

**NOTE.**  This command is applicable only for TDM audio bus type (BUS:B<x>AUDio:TYPE TDM)

**Conditions**  
This command requires DPO4AUDIO application module.

**Group**  Bus

**Syntax**  
BUS:B<x>:AUDio:CHANnel:SIZE <NR1>  
BUS:B<x>:AUDio:CHANnel:SIZE?

**Arguments**  
<NR1> specifies the number of bits per channel.

**Examples**  
BUS:B1:AUDIO:CHANNEL:SIZE 8 sets the number of bits per channel to 8.  

**BUS:B<x>:AUDio:CLOCk:POLarity**  
Sets or returns the clock polarity for the AUDIO bus.
**Conditions**
This command requires DPO4AUDIO application module.

**Group**
Bus

**Syntax**

```
BUS:B<x>:AUDio:CLOCk:POLarity {FALL|RISe}
BUS:B<x>:AUDio:CLOCk:POLarity?
```

**Arguments**

- 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:B1:AUDIO:CLOCK:POLARITY RISE indicating that the clock polarity
is set to Rise.
```

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

Sets or returns the clock source waveform for the AUDIO bus.

**Conditions**
This command requires DPO4AUDIO application module.

**Group**
Bus

**Syntax**

```
BUS:B<x>:AUDio:CLOCk:SOUrce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}
BUS:B<x>:AUDio:CLOCk:SOUrce?
```

**Arguments**

- CH1–Ch4 or D0–D15 specifies the clock source for the audio bus.

**Examples**

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

:BUS:B1:AUDIO:CLOCK:SOURCE CH1 indicating that the clock source is
set to CH1.
```
**BUS:B<x>:AUDio:DATa:POLarity**

Sets or returns the data polarity for the AUDIO bus.

**Conditions**
This command requires DPO4AUDIO application module.

**Group**
Bus

**Syntax**

```
BUS:B<x>:AUDio:DATa:POLarity {NORMal|INVERTed}
BUS:B<x>:AUDio:DATa:POLarity?
```

**Arguments**
- **NORMal** specifies positive data polarity for the audio bus.
- **INVERTed** specifies negative data polarity for the audio bus.

**Examples**
- `BUS:B1:AUDIO:DATA:POLARITY INVERTed` sets the data polarity to Inverted.

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

Sets or returns the number of bits per word for the AUDIO bus.

**NOTE.** The number of bits specified for this command must be always less than or equal to the number of bits per channel (`BUS:B<x>:AUDio:CHANnel:SIZe`).

**Conditions**
This command requires DPO4AUDIO application module.

**Group**
Bus

**Syntax**

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

**Arguments**
- **<NR1>** specifies the number of bits per word.
Examples

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

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

Sets or returns the data source waveform for the AUDIO bus.

**Conditions**

This command requires DPO4AUDIO application module.

**Group**

Bus

**Syntax**

```plaintext
BUS:B<x>:AUDio:DATa:SOUrce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}
BUS:B<x>:AUDio:DATa:SOUrce?
```

**Arguments**

CH1–Ch4 or D0–D15 specifies the channel to use for data source for the audio bus.

**Examples**

```plaintext
BUS:B1:AUDIO:DATA:SOURCE D1 sets the data source to D1.
```

**BUS:B<x>:AUDio:DISplay:FORMat**

Sets or returns the display format for the AUDIO bus.

**Conditions**

This command requires DPO4AUDIO application module.

**Group**

Bus

**Syntax**

```plaintext
BUS:B<x>:AUDio:DISplay:FORMat {BINary|HEXadecimal|SIGNEDDECimal}
BUS:B<x>:AUDio:DISplay:FORMat?
```

**Arguments**

BINary specifies a binary data display.

HEXadecimal specifies a hexadecimal data display.
SIGNEDDECIMAL specifies a signed decimal data display.

**Examples**

BUS:B1:AUDIO:DISPLAY:FORMAT BINARY sets the display format to Binary.


### BUS:B<x>:AUDio:FRAME:SIZE

Sets or returns the number of channels in each frame for the AUDIO bus.

**NOTE.** This command is applicable only for TDM audio bus type (BUS:B<x>AUDio:TYPe TDM)

**Conditions**

This command requires DPO4AUDIO application module.

**Group**

Bus

**Syntax**

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

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

**Arguments**

<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:FRAMESync:POLarity

Sets or returns the frame sync polarity for the AUDIO bus.

**Conditions**

This command requires DPO4AUDIO application module.

**Group**

Bus

**Syntax**

BUS:B<x>:AUDio:FRAMESync:POLarity {FALL|RISe}

BUS:B<x>:AUDio:FRAMESync:POLarity?
Arguments

FALL specifies the falling edge as the frame sync polarity.
RISe specifies the rising edge as the frame sync polarity.

Examples

BUS:B1:AUDIO:FRAMESYNC:POLARITY FALL sets the falling edge for frame sync polarity.
:BUS:B1:AUDIO:FRAMESYNC:POLARITY RISe indicating that the polarity is set to Rise.

BUS:B<x>:AUDIo:FRAMESync:SOUrce

Sets or returns the frame sync source waveform for the AUDIO bus.

Conditions

This command requires DPO4AUDIO application module.

Group

Bus

Syntax

BUS:B<x>:AUDIo:FRAMESync:SOUrce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}
BUS:B<x>:AUDIo:FRAMESync:SOUrce?

Arguments

CH1–CH4 or D0–D15 specifies the channel to use as the frame sync source.

Examples

BUS:B1:AUDIO:FRAMESYNC:SOURCE CH1 sets CH1 as the frame sync source.
:BUS:B1:AUDIO:FRAMESYNC:SOURCE Ch2 indicating that the source is set to Ch2.

BUS:B<x>:AUDIo:TYPe

Sets or returns the audio format (type) for the AUDIO bus.

Conditions

This command requires DPO4AUDIO application module.

Group

Bus
Syntax

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

Arguments

I2S specifies l'2S audio format.
LJ specifies left-justified audio format.
RJ specifies right justified audio format.
TDM specifies 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 I'2S.

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

Sets or returns the word select polarity for the AUDIO bus.

Conditions

This command requires DPO4AUDIO application module.

Group

Bus

Syntax

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

Arguments

NORMal specifies positive WORDSel polarity.
INVERTed specifies negative WORDSel polarity.

Examples

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

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

Sets or returns the word select source waveform for the AUDIO bus.

Conditions

This command requires DPO4AUDIO application module.
Commands Listed in Alphabetical Order

**Group**  
Bus

**Syntax**  
BUS:B<x>:AUDio:WORDSel:SOUrce {CH1|CH2|CH3|CH4 |D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}  
BUS:B<x>:AUDio:WORDSel:SOUrce?

**Arguments**  
CH1–CH4 specifies the channel to use as the word select source.  
D0–D15 specifies the digital channel to use as the word select source.

**Examples**  
BUS:B1:AUDIO:WORDSEL:SOURCE CH1 sets CH1 as the word select source.  

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

Sets or returns the bit rate for CAN bus <x>, where x is the bus number.

**Conditions**  
This command requires a DPO4AUTO application module.

**Group**  
Bus

**Syntax**  
BUS:B<x>:CAN:BITRate  
<NR1>|{RATE10K|RATE20K|RATE33K|RATE37K|RATE50K|RATE62K| RATE83K|RATE92K|RATE100K|RATE125K|RATE250K|RATE500K| RATE800K|RATE1M}  
BUS:B<x>:CAN:BITRate?

**Arguments**  
<NR1> is the bit rate. The instrument supports bit rates at 10 bps intervals. You can enter any positive integer, and the instrument will coerce the value to the closest supported bit rate.

**Returns**  
The query always returns the numerical bit rate value.

**Examples**  
bus:b1:can:bitrate rate400k sets the CAN bit rate to 400K.  
bus:b1:can:bitrate? might return :BUS:B1:CAN:BITRATE RATE800K indicating the bit rate is set to 800K.
**BUS:B<x>:CAN:PRObe**

Sets or returns the probing method to probe CAN bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4AUTO application module.

**Group**
Bus

**Syntax**
BUS:B<x>:CAN:PRObe {CANH|CANL|RX|TX|DIFFerential}
BUS:B<x>:CAN:PRObe?

**Arguments**
CANH specifies the single-ended CANH signal, as specified by the CAN standard.
CANL specifies the single-ended CANL signal, as specified by the CAN standard.
RX specifies the receive signal on the bus side of the CAN transceiver.
TX specifies the transmit signal.
DIFFerential specifies the differential CAN signal.

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

Sets or returns the sampling point during each bit period for bus <x>, where x is the bus number

**Conditions**
This command requires a DPO4AUTO application module.

**Group**
Bus

**Syntax**
BUS:B<x>:CAN:SAMPLEpoint <NR1>
BUS:B<x>:CAN:SAMPLEpoint?

**Arguments**
<NR1> is the sample point in percent. Values are limited to 25, 30, ... 70, 75.

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

Sets or returns the CAN bus data source for bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4AUTO application module.
Commands Listed in Alphabetical Order

Group | Bus
Syntax | BUS:B<x>:CAN:SOURc{CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15} BUS:B<x>:CAN:SOURc?

Arguments
- CH1–CH4 is the analog channel to use as the data source.
- D0–D15 is the digital channel to use as the data source.

BUS:B<x>:DISplay:FORMAT

Sets or returns the display format for the numerical information in the bus waveform, where <x> is the bus number. The display formats supported depend on the BUS:B<x>:TYPE.

Table 2-37: Supported display formats

<table>
<thead>
<tr>
<th>Bus type</th>
<th>Display format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel</td>
<td>BINary</td>
</tr>
<tr>
<td>I2C</td>
<td>BINary</td>
</tr>
<tr>
<td>SPI</td>
<td>BINary</td>
</tr>
<tr>
<td>CAN</td>
<td>BINary</td>
</tr>
<tr>
<td>RS232C</td>
<td>BINary</td>
</tr>
<tr>
<td>Audio</td>
<td>BINary</td>
</tr>
<tr>
<td>LIN</td>
<td>BINary</td>
</tr>
<tr>
<td>FlexRay</td>
<td>BINary</td>
</tr>
<tr>
<td>USB</td>
<td>BINary</td>
</tr>
</tbody>
</table>

† SIGNEDDECIMAL is set using the audio application BUS:B<x>:AUDio:DISplay:FORMAT command.

Conditions
- This command requires a serial bus analysis module.

Group | Bus
Syntax | BUS:B<x>:DISplay:FORMAT {BINary|HEXadecimal|ASCII|MIXed|MIXED|MIXED2} BUS:B<x>:DISplay:FORMAT?

Related Commands | BUS:B<x>:TYPE
Arguments

BINary – All values are displayed in binary.

HEXadecimal – All values are displayed in hexadecimal.

ASCII – All values are displayed in an ASCII format, for RS-232 only.

MIXed – Values are displayed in a mixture of hexadecimal, binary, and decimal, depending on the field.

MIXED2 – Values are displayed in a mixture of hexadecimal, binary, decimal and ASCII, depending on the field.

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

Sets or returns the display type for bus <x>, where x is the bus number. You can set up the bus to display the protocol information, the logic waveforms that comprise the bus, or both.

Conditions

This command requires a serial bus analysis module.

Group

Bus

Syntax

BUS:B<x>:DIsplay:TYPe {BUS|Both}

BUS:B<x>:DIsplay:TYPe?

Arguments

BUS displays the bus waveforms only.

Both displays both the bus and logic waveforms.

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

Sets or returns the bit rate for FLEXRAY.

Group

Bus

Syntax

BUS:B<x>:FLEXray:BITRate <NR1>

BUS:B<x>:FLEXray:BITRate?

Arguments

<NR1> specifies the FLEXRAY bit rate. You can enter any positive integer, and the instrument will coerce the value to the closest supported bit rate.
**Examples**  

BUS:B1:FLEXRAY:BITRATE 9600 sets the FLEXRAY bit rate to 9600 bits per second.  

BUS:B1:FLEXRAY:BITRATE? might return BUS:B1:FLEXRAY:BITRATE 10000000 indicating the FLEXRAY bit rate is 10,000,000 bits per second.

---

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

Sets or returns the FLEXRAY ID format.

**Group**  
Bus

**Syntax**  

BUS:B<x>:FLEXray:Channel {A|B}  
BUS:B<x>:FLEXray:Channel?

**Arguments**

A sets the FLEXRAY id format to channel A.  
B sets the FLEXRAY id format to channel B.

**Examples**

BUS:B1:FLEXRAY:CHANNEL B sets the FLEXRAY id format to channel B.  

BUS:B1:FLEXRAY:CHANNEL? might return BUS:B1:FLEXRAY:CHANNEL A indicating that FLEXRAY id format is channel A.

---

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

Sets or returns the FLEXRAY standard.

**Group**  
Bus

**Syntax**  

BUS:B<x>:FLEXray:Signal {BDIFFBP|BM|TXRX}  
BUS:B<x>:FLEXray:Signal?

**Arguments**

BDIFFBP sets the FLEXRAY standard to BDIFFBP.  
BM sets the FLEXRAY standard to BM.  
TXRX sets the FLEXRAY standard to TXRX.
Examples

BUS:B1:FLEXRAY:SIGNAL  BM sets the FLEXRAY standard is BM.

BUS:B1:FLEXRAY:SIGNAL? might return BUS:B1:FLEXRAY:SIGNAL BDIFFBP indicating the FLEXRAY standard is BDIFFBP.

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

Sets or returns the FLEXRAY data source.

**Group**

Bus

**Syntax**

BUS:B<x>:FLEXray:SOURce

{CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}

BUS:B<x>:FLEXray:SOURce?

**Arguments**

CH<x> sets the FLEXRAY source to channel x, where x is 1 to 4.

D<x> sets the FLEXRAY source to digital channel x, where x is 1 to 16.

**Examples**

BUS:B1:FLEXRAY:SOURce CH4 sets the FLEXRAY source to channel 4.


**BUS:B<x>:I2C:ADDRess:RWINClude**

Sets and returns whether the read/write bit is included in the address.

**Group**

Bus

**Syntax**

BUS:B<x>:I2C:ADDRess:RWINClude {<NR1>|OFF|ON}

BUS:B<x>:I2C:ADDRess:RWINClude?

**Arguments**

<NR1> = 0 does not include the read/write bit in the address; any other value includes the read/write bit in the address.

OFF does not include the read/write bit in the address.

ON includes the read/write bit in the address.

**Examples**

BUS:B1:I2C:ADDRess:RWINClude ON includes the read/write bit in the address.
BUS:B1:I2C:ADDRESS:RWINCLUDE? might return
BUS:B1:I2C:ADDRESS:RWINCLUDE 0 indicating the read/write bit is not included in the address.

**BUS:B<x>:I2C{:CLOCK|:SCLK}:SOUrce**

Sets or returns the I2C SCLK source for bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4EMBD or DPO4COMP application module.

**Group**
Bus

**Syntax**
BUS:B<x>:I2C{:CLOCK|:SCLK}:SOURce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}
BUS:B<x>:I2C{:CLOCK|:SCLK}:SOURce?

**Arguments**
CH1–CH4 specifies the analog channel to use as the I2C SCLK source.
D0–D15 specifies the digital channel to use as the I2C SCLK source.

**BUS:B<x>:I2C{:DATA|:SDATA}:SOUrce**

Sets or returns the I2C SDATA source for bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4EMBD application module.

**Group**
Bus

**Syntax**
BUS:B<x>:I2C{:DATA|:SDATA}:SOURce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}
BUS:B<x>:I2C{:DATA|:SDATA}:SOURce?

**Arguments**
CH1–CH4 specifies the analog channel to use as the I2C SDATA source.
D0–D15 specifies the digital channel to use as the I2C SCLK source.

**BUS:B<x>:LABel**

Sets or returns the waveform label for bus <x>, where x is the bus number 1 through 4.
Commands Listed in Alphabetical Order

**Group**  Bus

**Syntax**  
BUS:B<x>:LABel <Qstring>  
BUS:B<x>:LABel?

**Arguments**  
<Qstring> is an alphanumeric string of text, enclosed in quotes, that contains the text label information for bus <x>. The text string is limited to 30 characters.

**BUS:B<x>:LIN:BITRate**

Sets or returns the bit rate for LIN.

**Group**  Bus

**Syntax**  
BUS:B<x>:LIN:BITRate <NR1>  
BUS:B<x>:LIN:BITRate?

**Arguments**  
<NR1> is the LIN bit rate. You can enter any positive integer, and the instrument will coerce the value to the closest supported bit rate.

**Examples**  
BUS:B1:LIN:BITRATE 9600 sets the bit rate 9600.  

**BUS:B<x>:LIN:IDFORmat**

Sets or returns the LIN ID format.

**Group**  Bus

**Syntax**  
BUS:B<x>:LIN:IDFORmat {NOPARity|PARity}  
BUS:B<x>:LIN:IDFORmat?

**Arguments**  
NOPARity sets the LIN id format to no parity.  
PARity sets the LIN id format to parity.
Examples  
`BUS:B1:LIN:IDFORMAT PARITY` sets the LIN id format to parity.

**BUS:B<x>:LIN:POLARity**

Sets or returns the LIN polarity.

**Group**  
Bus

**Syntax**  
`BUS:B<x>:LIN:POLARity {NORMAL|INVERTED}`
`BUS:B<x>:LIN:POLARity?`

**Arguments**  
NORMAL specifies normal LIN polarity.
INVERTED specifies inverted LIN polarity.

**Examples**  
`BUS:B1:LIN:POLARITY INVERTED` sets the LIN polarity to INVERTED.

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

Sets or returns the sample point (in %) at which to sample during each bit period.

**Group**  
Bus

**Syntax**  
`BUS:B<x>:LIN:SAMPLEpoint <NR1>`
`BUS:B<x>:LIN:SAMPLEpoint?`

**Arguments**  
`<NR1>` is the sample point (in %) at which to sample during each bit period.

**Examples**  
`BUS:B1:LIN:SAMPLEPOINT 10` sets the sample point is at 10% of the bit period
`BUS:B1:LIN:SAMPLEPOINT?` might return `BUS:B1:LIN:SAMPLEPOINT 50` indicating that the sample point is at 50% of the bit period
**BUS:B<x>:LIN:SOURce**

Sets or returns the LIN data source.

**Group** Bus

**Syntax**

```
BUS:B<x>:LIN:SOURce
{CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}
BUS:B<x>:LIN:SOURce?
```

**Arguments**

- CH<x> specifies the LIN source channel where x is 1 to 4.
- D<x> specifies the LIN source channel where x is 1 to 16.

**Examples**

`BUS:B1:LIN:SOURce CH4` sets the LIN source to channel 4.

`BUS:B1:LIN:SOURce?` might return `BUS:B1:LIN:SOURce CH1` indicating the LIN source is channel 1.

**BUS:B<x>:LIN:STANDard**

Sets or returns the LIN standard.

**Group** Bus

**Syntax**

```
BUS:B<x>:LIN:STANDard {V1X|V2X|MIXed}
BUS:B<x>:LIN:STANDard?
```

**Arguments**

- V1X sets the LIN standard to V1X.
- V2X sets the LIN standard to V2X
- MIXed sets the LIN standard to MIXED.

**Examples**

`BUS:B1:LIN:STANDard V1X` sets the LIN standard is V1X.

`BUS:B1:LIN:STANDard?` might return `BUS:B1:LIN:STANDard V2X` indicating the LIN standard is V2X.
**BUS:B<x>:PARallel:BIT<x>:SOUrce**

Sets or returns the parallel bit source for B<x>, where x is the bus number and Bit<x> is the bit number.

**Group**  
Bus

**Syntax**  
BUS:B<x>:PARallel:BIT<x>:SOUrce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}  
BUS:B<x>:PARallel:BIT<x>:SOUrce?

**Arguments**  
CH1–CH4 or D0–D15 specifies the bit source B<x>. BIT<x> specifies the bit number.

**BUS:B<x>:PARallel:CLOCK:EDGE**

Sets or returns the parallel clock edge for bus <x>, where x is the bus number.

**Group**  
Bus

**Syntax**  
BUS:B<x>:PARallel:CLOCK:EDGE {EITher|RISing|FALling}  
BUS:B<x>:PARallel:CLOCK:EDGE?

**Arguments**  
EITher specifies either edge as the clock edge.  
RISing specifies the rising edge as the clock edge.  
FALling specifies the falling edge as the clock edge.

**BUS:B<x>:PARallel:CLOCK:ISCLOCKed**

Sets or returns the parallel bus clock function for bus <x>, where x is the bus number.

**Group**  
Bus

**Syntax**  
BUS:B<x>:PARallel:CLOCK:ISCLOCKed {YES|NO}  
BUS:B<x>:PARallel:CLOCK:ISCLOCKed?
Arguments YES specifies that the Parallel bus is clocked.

NO specifies that the Parallel bus is not clocked.

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

Sets or returns the Parallel bus source <x>, where x is the bus number.

**Group** Bus

**Syntax**

BUS:B<x>:PARallel:CLOCK:SOUrce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}

BUS:B<x>:PARallel:CLOCK:SOUrce?

**Arguments** CH1–CH4 or D0–D15 specifies the channel to use as the Parallel bit source.

**BUS:B<x>:PARallel:WIDth**

Sets or returns the number of bits used for the width of the Parallel bus <x>, where x is the bus number.

**Group** Bus

**Syntax**

BUS:B<x>:PARallel:WIDth <NR1>

BUS:B<x>:PARallel:WIDth?

**Arguments** <NR1> is the number of bits.

**BUS:B<x>:POSition**

Sets or returns the position of the bus <x> waveform on the display, where x is the bus number 1 through 4.

**Conditions** This command requires a serial bus analysis module.

**Group** Bus
Commands Listed in Alphabetical Order

Syntax

BUS:B<x>:POSITION <NR3>
BUS:B<x>:POSITION?

Arguments

<NR3> specifies the position.

BUS:B<x>:RS232C:BITRate

Sets or returns the RS-232 bit rate for bus <x>, where x is the bus number.

Conditions

This command requires a DPO4COMP application module.

Group

Bus

Syntax

BUS:B<x>:RS232C:BITRate <NR1>
BUS:B<x>:RS232C:BITRate?

Arguments

<NR1> is the bit rate in bits-per-second. You can enter any positive integer, and
the instrument will coerce the value to the closest supported bit rate.

BUS:B<x>:RS232C:DATABits

Sets or returns the number of RS-232 data bits for bus <x>, where x is the bus number.

Conditions

This command requires a DPO4COMP application module.

Group

Bus

Syntax

BUS:B<x>:RS232C:DATABits {7|8}
BUS:B<x>:RS232C:DATABits?

Arguments

7 specifies seven bits in the RS-232 data frame.
8 specifies eight bits in the RS-232 data frame.
**BUS:B<x>:RS232C:DELIMiter**

Sets or returns the RS-232 delimiting value for a packet on bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4COMP application module.

**Group**
Bus

**Syntax**

```
BUS:B<x>:RS232C:DELIMiter {NULl|LF|CR|SPace|XFF}
BUS:B<x>:RS232C:DELIMiter?
```

**Arguments**

- **NULl** specifies 0x00.
- **LF** specifies 0x0A.
- **CR** specifies 0x0D.
- **XFF** specifies 0xFF.

---

**BUS:B<x>:RS232C:DISPLAYmode**

Sets or returns the display mode for the bus <x> display and event table, where x is the bus number.

**Conditions**
This command requires a DPO4COMP application module.

**Group**
Bus

**Syntax**

```
BUS:B<x>:RS232C:DISPLAYmode {FRAme|PACKET}
BUS:B<x>:RS232C:DISPLAYmode?
```

**Arguments**

- **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 or the front panel.

---

**BUS:B<x>:RS232C:PARity**

Sets or returns the RS-232 parity for bus <x>, where x is the bus number.
### BUS:B<x>:RS232C:PARity

Sets or returns the RS-232C parity for bus \(<x>\), where \(x\) is the bus number.

**Syntax**

\[
\text{BUS:B}\langle x\rangle:RS232C:PARity \{\text{NONE} | \text{EVEN} | \text{ODD}\} \\
\text{BUS:B}\langle x\rangle:RS232C:PARity? 
\]

**Arguments**

- **NONE** specifies no parity.
- **EVEN** specifies even parity.
- **ODD** specifies odd parity.

### BUS:B<x>:RS232C:POLarity

Sets or returns the RS-232C polarity for bus \(<x>\), where \(x\) is the bus number.

**Syntax**

\[
\text{BUS:B}\langle x\rangle:RS232C:POLarity \{\text{NORMAL} | \text{INVERTed}\} \\
\text{BUS:B}\langle x\rangle:RS232C:POLarity? 
\]

**Arguments**

- **NORMAL** sets the RS-232C bus polarity to positive.
- **INVERTed** sets the RS-232C bus polarity to negative.

### BUS:B<x>:RS232C:RX:SOURce

Sets or returns the RS-232 RX source for bus \(<x>\), where \(x\) is the bus number.

**Syntax**

\[
\text{BUS:B}\langle x\rangle:RS232C:RX:SOURce \{\text{CH1} | \text{CH2} | \text{CH3} | \text{CH4} | \\
\text{D0} | \text{D1} | \text{D2} | \text{D3} | \text{D4} | \text{D5} | \text{D6} | \text{D7} | \text{D8} | \text{D9} | \text{D10} | \text{D11} | \text{D12} | \text{D13} | \text{D14} | \text{D15} | \text{Off}\} \\
\text{BUS:B}\langle x\rangle:RS232C:RX:SOURce? 
\]
**Arguments**  
CH1–CH4 or D0–D15 specifies the channel to use for the RS-232 RX source.  
Off sets the specified bus input to off.

**BUS:B<x>:RS232C:TX:SOUrce**

Sets or returns the RS-232 TX Source for bus <x>, where x is the bus number.

**Conditions**  
This command requires a DPO4COMP application module.

**Group**  
Bus

**Syntax**  
BUS:B<x>:RS232C:TX:SOUrce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15|Off}  
BUS:B<x>:RS232C:TX:SOUrce?

**Arguments**  
CH1–CH4 or D0–D15 specifies the channel to use as the RS-232 TX source.  
Off sets the specified bus input to off.

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

Sets or returns the bit order for the specified SPI bus. LSB is least significant bit first; MSB is most significant bit first.

**Group**  
Bus

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

**Arguments**  
LSB sets the bit order to least significant bit first.  
MSB sets the bit order to most significant bit first.

**Examples**  
BUS:B1:SPI:BITORDER LSB sets the bit order to least significant bit first.  
**BUS:B<x>:SPI{:CLOCK|:SCLK}:POLARity**

Sets or returns the SPI SCLK polarity for bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4EMBD application module.

**Group**
Bus

**Syntax**

```
BUS:B<x>:SPI{:CLOCK|:SCLK}:POLARity {FALL|RISe}
BUS:B<x>:SPI{:CLOCK|:SCLK}:POLARity?
```

**Arguments**
- FALL specifies the falling edge.
- RISE specifies the rising edge.

**BUS:B<x>:SPI{:CLOCK|:SCLK}:SOUrce**

Sets or returns the SPI SCLK source for bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4EMBD application module.

**Group**
Bus

**Syntax**

```
BUS:B<x>:SPI{:CLOCK|:SCLK}:SOUrce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}
BUS:B<x>:SPI{:CLOCK|:SCLK}:SOUrce?
```

**Arguments**
- CH1-CH4 or D0-D15 is the channel to use as the SPI SCLK source.

**BUS:B<x>:SPI:DATA{:IN|:MISO}:POLARity**

Sets or returns the SPI MISO polarity for bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4EMBD application module.

**Group**
Bus
**Syntax**

BUS:B<x>:SPI:DATA{:IN|:MISO}:POLARity {LOW|HIGH}
BUS:B<x>:SPI:DATA{:IN|:MISO}:POLARity?

**Arguments**

LOW specifies an active low polarity.

HIGH specifies an active high polarity.

**BUS:B<x>:SPI:DATA{:IN|:MISO}:SOUrce**

Sets or returns the SPI MISO source for bus <x>, where x is the bus number.

**Conditions**

This command requires a DPO4EMBD application module.

**Group**

Bus

**Syntax**

BUS:B<x>:SPI:DATA{:IN|:MISO}:SOUrce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}
BUS:B<x>:SPI:DATA{:IN|:MISO}:SOUrce?

**Arguments**

CH1–CH4 or D0–D15 is the channel to use as the SPI MISO source.

**BUS:B<x>:SPI:DATA{:OUT|:MOSI}:POLARity**

Sets or returns the SPI MOSI polarity for bus <x>, where x is the bus number.

**Conditions**

This command requires a DPO4EMBD application module.

**Group**

Bus

**Syntax**

BUS:B<x>:SPI:DATA{:OUT|:MOSI}:POLARity {LOW|HIGH}
BUS:B<x>:SPI:DATA{:OUT|:MOSI}:POLARity?

**Arguments**

LOW specifies the active low polarity.

HIGH specifies the active high polarity.

**BUS:B<x>:SPI:DATA{:OUT|:MOSI}:SOUrce**

Sets or returns the SPI MOSI source for bus <x>, where x is the bus number.
Conditions

This command requires a DPO4EMBD application module.

Group

Bus

Syntax

BUS:B<x>:SPI:DATA{:OUT|:MOSI}:SOURce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}

BUS:B<x>:SPI:DATA{:OUT|:MOSI}:SOURce?

Arguments

CH1-CH4 or D0-D15 is the channel to use as the SPI MISO source.

**BUS:B<x>:SPI:DATA:SIZE**

Sets or returns the number of bits per word for the specified SPI bus.

Group

Bus

Syntax

BUS:B<x>:SPI:DATA:SIZE <NR1>

BUS:B<x>:SPI:DATA:SIZE?

Arguments

NR1 is the data size of the specified bus.

Examples

BUS:B1:SPI:DATA:SIZE 8 sets the data size for the specified bus to 8 bits per word.


**BUS:B<x>:SPI:FRAMING**

Sets or returns the type of SPI framing.

Group

Bus

Syntax

BUS:B<x>:SPI:FRAMING {SS|IDLEtime}

BUS:B<x>:SPI:FRAMING?

Arguments

SS specifies framing by SS (non 2-wire).

IDLEtime specifies framing by Idle Time (2-wire).
Commands Listed in Alphabetical Order

**Examples**

BUS:B1:SPI:FRAMING SS sets the SPI framing type to SS.

BUS:B1:SPI:FRAMING IDELTIME might return BUS:B1:SPI:FRAMING IDELTIME indicating the SPI framing type is set to IDLETIME.

**BUS:B<x>:SPI:IDLETime**

Sets or returns the SPI bus idle time in seconds for the specified bus.

**Group**  
Bus

**Syntax**

BUS:B<x>:SPI:IDLETime <NR3>
BUS:B<x>:SPI:IDLETime?

**Arguments**

NR3 is the SPI bus idle time in seconds.

**Examples**

BUS:B1:SPI:IDLETIME 100.0000E-9 sets the idle time to 100 ns.

BUS:B1:SPI:IDLETIME? might return BUS:B1:SPI:IDLETIME 100.0000E-9 indicating the idle time is set to 100 ns.

**BUS:B<x>:SPI{:SELect|:SS}:POLARity**

Sets or returns the SPI SS polarity for bus <x>, where x is the bus number.

**Conditions**

This command requires a DPO4EMBD application module.

**Group**  
Bus

**Syntax**

BUS:B<x>:SPI{:SELect|:SS}:POLARity {LOW|HIGH}
BUS:B<x>:SPI{:SELect|:SS}:POLARity?

**Arguments**

LOW specifies an active low polarity.

HIGH specifies an active high polarity.

**BUS:B<x>:SPI{:SELect|:SS}:SOUrce**

Sets or returns the SPI SS source for bus <x>, where x is the bus number.
Conditions
This command requires a DPO4EMBD application module.

Group
Bus

Syntax
BUS:B<x>:SPI{:SELECT|:SS}:SOURCE {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}
BUS:B<x>:SPI{:SELECT|:SS}:SOURCE?

Arguments
CH1–CH4 or D0–D15 is the channel to use as the SPI SS source.

BUS:B<x>:STATE
Sets or returns the on/off state of bus <x>, where x is the bus number.

Group
Bus

Syntax
BUS:B<x>:STATE {<NR1>|OFF|ON}
BUS:B<x>:STATE?

Related Commands
SElEct:BUS<x>

Arguments
ON or <NR1> ≠ 0 turns on the bus.
OFF or <NR1> = 0 turns off the bus.

BUS:B<x>:TYPE
Sets or returns the bus type for <x>, where x is the bus number. The supported bus types are dependent on the oscilloscope model and the installed application keys.

Group
Bus

Syntax
BUS:B<x>:TYPE {I2C|SPI|CAN|RS232C|PARallel|USB}
BUS:B<x>:TYPE?

Arguments
I2C specifies the Inter-IC bus.
SPI specifies the Serial Peripheral Interface bus (not available on two-channel models).
CAN specifies the Controller Area Network bus.
RS232C specifies the RS-232C bus.
PARallel specifies the Parallel bus.
USB specifies the USB bus.

**BUS:B<x>:USB:BITRate**

Sets or returns the bit rate for the USB signal.

**Group**  
Bus

**Syntax**  
BUS:B<x>:USB:BITRate \{LOW|FULL|HIGH\}  
BUS:B<x>:USB:BITRate?

**Arguments**  
LOW indicates the bit rate is 1.5 Mbps.  
FULL indicates the bit rate is 12 Mbps.  
HIGH indicates the bit rate is 480 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, which indicates that the bit rate is 1.5 Mbps.

**BUS:B<x>:USB:PRObe**

Sets or returns the type of probe connected to the USB signal.

**Group**  
Bus

**Syntax**  
BUS:B<x>:USB:PRObe \{DIFFerential|SINGleended\}  
BUS:B<x>:USB:PRObe?

**Arguments**  
DIFFerential indicates that the bus probe is a differential probe.  
SINGleended indicates that the bus probe is not a differential probe.

**Examples**  
BUS:B1:USB:PROBE DIFFERENTIAL sets the bus probe to be a differential probe.

### BUS:B<x>:USB:SOURce:DIFFerential

Sets or returns the USB Data Source when using a differential probe.

**Group**  
Bus

**Syntax**  
BUS:B<x>:USB:SOURce:DIFFerential {CH1|CH2|CH3|CH4}  
BUS:B<x>:USB:SOURce:DIFFerential?

**Arguments**  
CH1–CH4 specifies the channel to use for data source for the USB bus. This channel should have an attached differential probe.

**Examples**  
BUS:B1:USB:SOURce:DIFFerential CH2 sets the source to channel 2. You should have a differential probe attached to channel 2 and connect it to the USB data signal.


### BUS:B<x>:USB:SOURce:DMINus

Sets or returns the USB Data Source for D- input. If you are using single-ended probes, you need to set the sources for both the D+ and D- inputs.

**Group**  
Bus

**Syntax**  
BUS:B<x>:USB:SOURce:DMINus {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}  
BUS:B<x>:USB:SOURce:DMINus?

**Arguments**  
CH1–CH4 or D0–D15 specifies the channel to use for D- data source for the USB bus.
**Examples**


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

Sets or returns the USB Data Source for D+ input. If you are using single-ended probes, you need to set the sources for both the D+ and D- inputs.

**Group**

Bus

**Syntax**

`BUS:B<x>:USB:SOURce:DPLUs {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}`

**Arguments**

CH1–CH4 or D0–D15 specifies the channel to use for D+ data source for the USB bus.

**Examples**

`BUS:B1:USB:SOURce:DPLUs CH1` sets the D+ source to channel 1.

`BUS:B1:USB:SOURce:DPLUs?` might return `:BUS:B1:USB:SOURce:DPLUs CH3` indicating that channel 3 is set to be the D+ input for USB data.

**BUS:LOWerthreshold:CH<x>**

Sets the lower threshold for each channel. This applies to all search and trigger types that use the channel. This command supersedes the `:BUS:THReshold:CH<x>` above.

**Group**

Bus

**Syntax**

`BUS:LOWerthreshold:CH<x> {<NR3>|ECL|TTL}`

`BUS:LOWerthreshold:CH<x>?`

**Arguments**

`<NR3>` specifies the threshold in volts.

ECL specifies a preset ECL high level of -1.3V.

TTL specifies a preset TTL high level of 1.4V.
Examples

BUS:LOWERTHRESHOLD:CH1 TTL sets the CH1 lower threshold to 800mV.

BUS:LOWERTHRESHOLD:CH1? might return:BUS:LOWERTHRESHOLD:CH1 -800.0000E-3 indicating the CH1 lower threshold is -800 mV.

**BUS:THReshold:CH<x>**

Sets or returns the threshold for analog channel <x>, where x is the channel number. This setting applies to all trigger types that use the channel.

**Conditions**

This command requires a serial bus analysis module.

**Group**

Bus

**Syntax**

BUS:THReshold:CH<x> {ECL|TTL|<NR3>}

BUS:THReshold:CH<x>?  

**Related Commands**

TRIGger:A:LEVel:CH<x>

**Arguments**

ECL specifies a preset ECL high level of –1.3V.

TTL specifies a TTL preset high level of 1.4V.

<NR3> specifies the threshold level in volts.

**BUS:THReshold:D<x>**

Sets or returns the threshold for digital channel <x>, where x is the digital channel number. This will apply to all Search and Trigger Types that use the channel.

**Conditions**

This command requires a DPO4COMP application module or RS-232C.

**Group**

Bus

**Syntax**

BUS:THReshold:D<x> {<NR3>|ECL|TTL}

BUS:THReshold:D<x>?  

**Related Commands**

TRIGger:A:LEVel:D<x>
Arguments

ECL specifies a preset ECL high level of -1.3V.
TTL specifies a preset TTL high level of 1.4V.
<NR3> specifies the threshold level in volts.

**BUS:UPPerthreshold:**CH<x>

Sets the upper threshold for each channel. This applies to all search and trigger types that use the channel.

**Group**  Bus

**Syntax**  

BUS:UPPerthreshold:CH<x> {<NR3>|ECL|TTL}  
BUS:UPPerthreshold:CH<x>?  

**Arguments**  

<NR3> specifies the threshold in volts.
ECL specifies a preset ECL high level of -1.3V.
TTL specifies a preset TTL high level of 1.4V.

**Examples**  

BUS:UPPERTHRESHOLD:CH1 800.0000E-3 sets the CH1 upper threshold to 800 mV.
BUS:UPPERTHRESHOLD:CH1? might return :BUS:UPPERTHRESHOLD:CH1 -800.0000E-3 indicating that the CH1 upper threshold is set to -800 mV.

**BUSY? (Query Only)**

Returns the status of the oscilloscope. This command allows you to synchronize the operation of the oscilloscope with your application program. (See page 3-7, *Synchronization Methods*.)

Certain oscilloscope operations can affect the BUSY? response. (See Table 3-3 on page 3-7.)

**Group**  Status and Error

**Syntax**  BUSY?

**Related Commands**  *OPC, *WAI
**Returns**

<NR1> = 0 means the oscilloscope is not busy processing a command whose execution time is extensive.

<NR1> = 1 means the oscilloscope is busy processing a command whose execution time is extensive. (See Table 3-3 on page 3-7.)

**Examples**

BUSY? might return :BUSY 1 indicating that the oscilloscope is currently busy.

**CAL? (Query Only)**

Performs an internal self-calibration and returns the oscilloscope calibration status.

---

**NOTE.** Disconnect or otherwise remove all input signals prior to starting self-calibration. The self-calibration can take several minutes to complete.

No other commands are executed until calibration is complete.

**Group**

Calibration and Diagnostic

**Syntax**

*CAL?

**Returns**

<NR1> = 1 indicates the calibration did not complete successfully.

<NR1> = 0 indicates the calibration completed without errors.

**Examples**

*CAL? starts the internal signal path calibration and might return 0 to indicate that the calibration was successful.

**CALibrate:FACTory:STATus? (Query Only)**

Returns the factory calibration status value saved in nonvolatile memory.

**Group**

Calibration and Diagnostic

**Syntax**

CALibrate:FACTory:STATus?

**Examples**

**CALibrate:INTERNal (No Query Form)**

This command starts a signal path compensation.

**Group**  
Calibration and Diagnostic

**Syntax**  
CALibrate:INTERNal

**Arguments**  
None

**Examples**  
CALIBRATE:INTERNAL starts a serial path compensation cycle.

**CALibrate:INTERNal:STARt (No Query Form)**

Starts the internal signal path calibration (SPC) of the oscilloscope. You can use the CALibrate:INTERNal:STATus? query to return the current status of the internal signal path calibration of the oscilloscope.

**Group**  
Calibration and Diagnostic

**Syntax**  
CALibrate:INTERNal:STARt

**Related Commands**  
CALibrate:RESULTS:SPC?

**Examples**  
CALIBRATE:INTERNAL:START initiates the internal signal path calibration of the oscilloscope.

**CALibrate:INTERNal:STATus? (Query Only)**

Returns the current status of the oscilloscope internal signal path calibration for the last SPC operation.

**Group**  
Calibration and Diagnostic

**Syntax**  
CALibrate:INTERNal:STATus?

**Related Commands**  
*CAL?
Returns  This query will return one of the following:

- INIT indicates the oscilloscope has not had internal signal path calibration run.
- PASS indicates the signal path calibration completed successfully.
- FAIL indicates the signal path calibration did not complete successfully.
- RUNNING indicates the signal path calibration is currently running.

Examples  CALIBRATE:INTERNAL:STATUS? might return
CALIBRATE:INTERNAL:STATUS INIT indicating that the current status of the internal signal path calibration is that it has not been run.

CALibrate:RESults? (Query Only)

Returns the status of internal and factory calibrations, without performing any calibration operations. The results returned do not include the calibration status of attached probes. The CALibrate:RESults? query is intended to support GO/NoGO testing of the oscilloscope calibration readiness: all returned results should indicate PASS status if the oscilloscope is "fit for duty". It is quite common, however, to use uncalibrated probes (particularly when the oscilloscope inputs are connected into a test system with coaxial cables).

Group  Calibration and Diagnostic

Syntax  CALibrate:RESults?

Related Commands  *CAL?

CALibrate:RESults:FACTory? (Query Only)

Returns the status of internal and factory calibration, without performing any calibration operations.

Group  Calibration and Diagnostic

Syntax  CALibrate:RESults:FACTory?
**CALibrate:RES ults:SPC? (Query Only)**

Returns the status of the SPC operation. This query does not initiate a SPC.

**Group**  
Calibration and Diagnostic

**Syntax**  
CALibrate:RES ults:SPC?

**Related Commands**  
*CAL?

**Returns**  
INIT indicates that SPC has never successfully completed.  
PASS indicates that the last SPC operation passed.  
FAIL indicates that the last SPC operation failed.  
RUNNING indicates that the SPC operation is running.

**Examples**  
CALIBRATE:RESULTS:SPC? returns the results of the last SPC operation: either PASS or FAIL.

**CH<x>? (Query Only)**

Returns the vertical parameters for channel <x>, where x is the channel number.

**Group**  
Vertical

**Syntax**  
CH<x>?

**CH<x>:AMPSVIAVOLts:ENAble**

Sets or returns the state of the amps via volts feature for the specified channel. This feature supports measuring current via the voltage drop across a resistor.

**Group**  
Vertical

**Syntax**  
CH<x>:AMPSVIAVOLts:ENAble {<NR1>|OFF|ON}  
CH<x>:AMPSVIAVOLts:ENAble?
Arguments

- **OFF** sets the amps via volts function for channel <x> to off.
- **ON** sets the amps via volts function for channel <x> to on.
- \(<NR1> = 0\) sets the amps via volts function to off. Any other value sets the function to on.

Examples

```
CH1:AMSVIAVOLTS:ENABLE ON turns on the amps via volts feature for channel 1.
```

**CH<x>:AMPSVIAVOLTS:FACtor**

Sets or returns the amps via volts factor for the specified channel.

Group  
Vertical

Syntax

```
CH<x>:AMPSVIAVOLTS:FACtor <NR3>
CH<x>:AMPSVIAVOLTS:FACtor?
```

Arguments

- \(<NR3>\) is a double-precision ASCII string that represents the amps via volts factor.

Examples

```
CH1:AMPSVIAVOLTS:FACTOR 15.5 sets the amps via volts factor for channel 1 to 15.5.
```

**CH<x>:BANdwidth**

Sets or returns the selectable low-pass bandwidth limit filter for channel <x>, where x is the channel number.

Group  
Vertical

Syntax

```
CH<x>:BANdwidth {TWEnty|TWOfifty|FULl|<NR3>}
CH<x>:BANdwidth?
```

Arguments

- **TWEnty** sets the upper bandwidth limit of channel <x> to 20 MHz.
- **TWOfifty** sets the upper bandwidth limit of channel <x> to 250 MHz.
- **FULl** disables any optional bandwidth limiting. The specified channel operates at its maximum attainable bandwidth.
<NR3> is a double-precision ASCII string. The oscilloscope rounds this value to an available bandwidth using geometric rounding, and then uses this value to set the upper bandwidth limit.

**NOTE.** Other values may be possible depending on the attached probes.

### Examples

CH1:BANDWIDTH TWENTY sets the bandwidth of channel 1 to 20 MHz.

#### CH<x>:COUPling

Sets or returns the input attenuator coupling setting for channel <x>, where x is the channel number.

**Group** Vertical

**Syntax**

CH<x>:COUPling {AC|DC|GND}  
CH<x>:COUPling?

**Arguments**

AC sets channel <x> to AC coupling.

DC sets channel <x> to DC coupling.

GND sets channel<x> to ground. Only a flat, ground-level waveform will be displayed.

**Examples**

CH2:COUPling GND sets channel 2 to ground.

CH3:COUPling? might return :CH3:COUPling DC indicating that channel 3 is set to DC coupling.

#### CH<x>:DESKew

Sets or returns the deskew time for channel <x>, where x is the channel number. You can adjust the deskew time to add an independent, channel-based delay time to the delay (set by the horizontal position control and common to all channels) from the common trigger point to first sample taken for each channel. This lets you compensate individual channels for different delays introduced by their individual input hook ups.

**Group** Vertical
Commands Listed in Alphabetical Order

**Syntax**

CH<x>:DESKew <NR3>

CH<x>:DESKew?

**Arguments**

<NR3> is the deskew time for channel <x>, ranging from -100 ns to +100 ns with a resolution of 1 ps.

**Examples**

CH4:DESKew 5.0E-9 sets the deskew time for channel 4 to 5 ns.

CH2:DESKew? might return :CH2:DESKew 2.0000E-09 indicating that the deskew time for channel 2 is set to 2 ns.

---

**CH<x>:INVert**

Sets or returns the invert function for channel <x>, where <x> is the channel number. When on, the invert function inverts the waveform for the specified channel.

**NOTE.** This command inverts the waveform for display purposes only. The oscilloscope does not use an inverted waveform for triggers or trigger logic inputs.

**Group**

Vertical

**Syntax**

CH<x>:INVert {ON|OFF}

CH<x>:INVert?

**Arguments**

OFF sets the invert function for channel <x> to off.

ON sets the invert function for channel <x> to on.

**Examples**

CH4:INVert ON inverts the waveform on channel 4.

CH2:INVert? might return :CH2:INVERT 0 indicating that channel 2 is not inverted.

---

**CH<x>:LABel**

Sets or returns the waveform label for channel <x>, where x is the channel number.

**Group**

Vertical
Commands Listed in Alphabetical Order

**Syntax**

CH<x>:LABel <Qstring>
CH<x>:LABel?

**Arguments**

<Qstring> is an alphanumeric string of text, enclosed in quotes, that contains the
text label information for the channel <x> waveform. The text string is limited
to 30 characters.

**CH<x>:OFFSet**

Sets or returns the vertical offset for channel <x>, where x is the channel number.

This command offsets the vertical acquisition window (moves the level at the vertical
center of the acquisition window) for the specified channel. Visualize offset as scrolling
the acquisition window towards the top of a large signal for increased offset values,
and scrolling towards the bottom for decreased offset values. The resolution of the
vertical window sets the offset increment for this control.

Offset adjusts only the vertical center of the acquisition window for channel
waveforms to help determine what data is acquired. The oscilloscope always displays
the input signal minus the offset value.

The channel offset range depends on the vertical scale factor.

**Table 2-38: Channel Offset Range**

<table>
<thead>
<tr>
<th>V/Div Setting</th>
<th>Offset range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 MΩ Input</td>
</tr>
<tr>
<td>1 mV/div — 50 mV/div</td>
<td>±1 V</td>
</tr>
<tr>
<td>50.5 mV/div — 99.5 mV/div</td>
<td>±0.5 V</td>
</tr>
<tr>
<td>100 mV/div — 500 mV/div</td>
<td>±10 V</td>
</tr>
<tr>
<td>505 mV/div — 995 mV/div</td>
<td>±5 V</td>
</tr>
<tr>
<td>1 V/div — 5 V/div</td>
<td>±100 V</td>
</tr>
<tr>
<td>5.05 V/div — 10 V/div</td>
<td>±50 V</td>
</tr>
</tbody>
</table>

Footnote: For 50 Ω input, 1 V/div is the maximum setting.

**NOTE.** The above table describes oscilloscope behavior only when no probe is
attached, and when the external attenuation factor is 1.0.

**Group**

Vertical
Commands Listed in Alphabetical Order

Syntax

CH<x>:OFFSet <NR3>
CH<x>:OFFSet?

Related Commands

CH<x>:POSition

Arguments

<NR3> is the offset value for the specified channel <x>.

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

Sets or returns the vertical position of channel <x>, where x is the channel number. The position value is applied to the signal before it is digitized.

Increasing the position value of a waveform causes the waveform to move up. Decreasing the position value causes the waveform to move down. The position value determines the vertical graticule coordinate at which input signal values, minus the present offset setting for that channel, are displayed. For example, if the position for Channel 3 is set to 2.0 and the offset is set to 3.0, then input signals equal to 3.0 units are displayed 2.0 divisions above the center of the screen (at 1 V/div).

Group

Vertical

Syntax

CH<x>:POSition <NR3>
CH<x>:POSition?

Related Commands

CH<x>:OFFSet, REF<x>:VERTical:POSition, MATH[1]:VERTical:POSition

Arguments

<NR3> is the position value for channel <x>, in divisions, from the center graticule. The range is 8 to -8 divisions.

Examples

CH2:POSITION 1.3 positions the Channel 2 input signal 1.3 divisions above the center graticule.
CH1:POSITION? might return :CH1:POSITION -1.3000 indicating that the current position of Channel 1 is 1.3 divisions below the center graticule.
**CH<x>:PRObe? (Query Only)**

Returns all information concerning the probe attached to channel <x>, where x is the channel number.

**Group** Vertical

**Syntax** CH<x>:PRObe?

**CH<x>:PRObe:AUTOZero (No Query Form)**

Sets the TekVPI probe attached to channel <x> to zero, where x is the channel number

**Group** Vertical

**Syntax** CH<x>:PRObe:AUTOZero EXECute

**Arguments** Execute auto zeros the probe.

**CH<x>:PRObe:COMMAND (No Query Form)**

Sets the state of the probe control specified with the first argument to the state specified with the second argument. The commands and states are unique to the attached probe type. Only certain VPI probes support this command. See the probe documentation for how to set these string arguments.

**Group** Vertical

**Syntax** CH<x>:PRObe:COMMAND <QString>, <QString>

**Arguments**<QString> are quoted strings specifying the probe command and value to set in the probe attached to the specified channel.

**Examples**

CH1:PROBE:COMMAND "MODE", "4–4V1MHz" sets a Tektronix VPI-DPG probe to the 4-4V1MHz mode.

CH1:PROBE:COMMAND "OUTPUT", "ON" turns the output of a Tektronix VPI-DPG probe on.
CH1:PROBE:COMMAND?“MODE” might return CH1:PROBE:COMMAND “MODE”, “4-4V1MHZ”.

**CH<x>:PRObe:DEGAUss (No Query Form)**

Starts a degauss auto-zero cycle on a TekVPI current probe attached to the input channel specified by <x>, where x is the channel number.

**Group** Vertical

**Syntax** CH<x>:PRObe:DEGAUss EXECute

**Arguments** EXECute initiates the degauss operation.

**CH<x>:PRObe:DEGAUss:STATE? (Query Only)**

Returns the state of the probe degauss for the channel specified by <x>, where x is the channel number.

**NOTE.** This command will return PASSED for probes that do not support degauss operations.

**Group** Vertical

**Syntax** CH<x>:PRObe:DEGAUss:STATE?

**Returns** NEEDED 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. RUNNING indicates the probe degauss operation is currently in progress.

**CH<x>:PRObe:FORCEDRange**

Sets or returns the range of a TekVPI probe attached to the channel specified by <x>, where x is the channel number.
Commands Listed in Alphabetical Order

**NOTE.** This command will return *PASSED* for probes that do not support degauss operations.

<table>
<thead>
<tr>
<th>Group</th>
<th>Vertical</th>
</tr>
</thead>
</table>

**Syntax**

CH<x>:PRObe:FORCEDRange <NR3>
CH<x>:PRObe:FORCEDRange?

**Arguments**

<NR3> specifies the range, which is probe specific.

**Returns**

This command returns 0.0 for probes that do not support forced range.

### CH<x>:PRObe:GAIN

Sets or returns the gain factor for the probe attached to the channel specified by <x>, where x is the channel number. The "gain" of a probe is the output divided by the input transfer ratio. For example, a common 10x probe has a gain of 1V.

<table>
<thead>
<tr>
<th>Group</th>
<th>Vertical</th>
</tr>
</thead>
</table>

**Syntax**

CH<x>:PRObe:GAIN <NR3>
CH<x>:PRObe:GAIN?

**Related Commands**

CH<x>:SCA1e

**Arguments**

<NR3> is the probe gain. Allowed values depend on the specific probe.

**Examples**

CH2:PROBE:GAIN? might return :CH2:PROBE:GAIN 0.1000E+00 indicating that the attached 10x probe delivers 1 V to the channel 2 BNC for every 10 V applied to the probe input.

### CH<x>:PRObe:ID? (Query Only)

Returns the type and serial number of the probe attached to channel <x>, where x is the channel number.

| Group | Vertical |
Syntax  
\texttt{CH<x>:PRObe:ID?}

Examples  
\texttt{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.

\textbf{CH<x>:PRObe:ID:SERNUMBER? (Query Only)}

Returns the serial number of the probe attached to channel \(<x>\), where x is the channel number.

\textit{NOTE. For Level 0 and 1 probes, the serial number will be ".".}

Group  
Vertical

Syntax  
\texttt{CH<x>:PRObe:ID:SERNumber?}

Examples  
\texttt{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.

\textbf{CH<x>:PRObe:ID:TYPE? (Query Only)}

Returns the type of probe attached to the channel specified by \(<x>\), where x is the channel number. Level 2 (or higher) probes supply their exact product nomenclature; for Level 0 or 1 probes, a generic “No Probe Detected" message is returned.

Group  
Vertical

Syntax  
\texttt{CH<x>:PRObe:ID:TYPE?}

Examples  
\texttt{CH1:PROBE:ID:TYPE?} might return :CH1:PROBE:ID:TYPE "P6203" indicating that P6203-type probe is attached to channel 1.
**CH<x>:PRObe:MODEl**

Sets or returns the probe model for the specified channel. If a coded probe is attached or the specified model is not recognized, an error event is set. The argument must be a supported probe.

To find the list of supported probes, push the front-panel channel 1, 2, 3, or 4 button, then push the lower-menu More item as many times as needed to select Deskew. Push Probe Model on the side menu and turn knob a to help you read from the resulting list. Alternatively, you can push Test > Analysis > Deskew > Configure > Probe Model and read from the resulting list.

**Group** Vertical

**Syntax**

CH<x>:PRObe:MODEL<String>
CH<x>:PRObe:MODEl?

**CH<x>:PRObe:PROPDELay**

Sets or returns the propagation delay for the probe connected to the specified channel.

**Group** Vertical

**Syntax**

CH<x>:PRObe:PROPDELay<NR3>
CH<x>:PRObe:PROPDELay?

**Arguments** <NR3> specifies the propagation time delay for the connected probe.

**Examples**

CH1:PROBE:PROPDELAY 100E-12 sets the CH1 propagation delay to 100 ps.

CH1:PROBE:PROPDELAY? might return :CH1:PROBE:PROPDELAY 1.1E-9 indicating that the CH1 propagation delay is set to 1.1 ns.

**CH<x>:PRObe:RECDESkew? (Query Only)**

Returns the recommended deskew for the probe connected to the specified channel.

**Group** Vertical
**Syntax**  
`CH<x>:PRObe:RECDESkew?`

**CH<x>:PRObe:RESistance? (Query Only)**

Returns the resistance factor of the probe attached to channel `<x>`, where x is the channel number.

**Group**  
Vertical

**Syntax**  
`CH<x>:PRObe:RESistance?`

**Examples**  
`CH2:PRObe:RESistance?` might return `:CH2:PROBE:RESISTANCE 10.0000E+6` indicating that the input resistance of the probe attached to Channel 2 is 1 MΩ.

**NOTE.** This query will return 0.0 if no probe is attached or the attached probe does not report the input resistance.

**CH<x>:PRObe:SIGnal**

Sets or returns the input bypass setting of a TekVPI probe attached to channel `<x>`, where x is the channel number. The probe must support input bypass, for example TCP0001. This command is ignored if sent to an unsupported probe.

**Group**  
Vertical

**Syntax**  
`CH<x>:PRObe:SIGnal {BYPass|PASS}`  
`CH<x>:PRObe:SIGnal?`

**Arguments**  
BYPass sets the probe to Bypass mode.  
PASS sets the probe to Pass mode.

**CH<x>:PRObe:UNIts? (Query Only)**

Returns a string describing the units of measure for the probe attached to channel `<x>`, where x is the channel number.

**Group**  
Vertical
Commands Listed in Alphabetical Order

Syntax

\texttt{CH<x>:PRObe:UNIts?}

Examples

\texttt{CH4:PROBE:UNITS?} might return \texttt{:CH4:PROBE:UNITS "V"} indicating that the units of measure for the probe attached to channel 4 are volts.

**CH<x>:SCAle**

Sets or returns the vertical scale for the channel specified by \(<x>\), where \(x\) is the channel number.

Each waveform has a vertical scale parameter. For a signal with constant amplitude, increasing the Scale causes the waveform to be displayed smaller. Decreasing the scale causes the waveform to be displayed larger.

Scale affects all waveforms, but affects channel waveforms differently from other waveforms:

- For channel waveforms, this setting controls the vertical size of the acquisition window as well as the display scale. The range and resolution of scale values depends on the probe attached and any other external factors you have specified.

- For reference and math waveforms, this setting controls the display only, graphically scaling these waveforms and having no affect on the acquisition hardware.

Group

Vertical

Syntax

\texttt{CH<x>:SCAle <NR3>}  
\texttt{CH<x>:SCAle?}

Related Commands

\texttt{CH<x>:OFFSet, CH<x>:POSition, REF<x>:VERTical:SCAle, MATH[1]:VERTical:SCAle}

Arguments

\(<NR3>\) is the vertical channel scale in units-per-division. The value entered here is truncated to three significant digits.

Examples

\texttt{CH4:SCALE 100E-03} sets the channel 4 scale to 100 mV per division.

\texttt{CH2:SCALE?} might return \texttt{:CH2:SCALE 1.0000} indicating that the current scale setting of channel 2 is 1 V per division.
**CH<x>:TERmination**

Sets the connected-disconnected status of a 50 Ω resistor, which may be connected between the specified channel’s coupled input and oscilloscope ground. The channel is specified by <x>. There is also a corresponding query that requests the termination parameter and translates this enumeration into one of the two float values.

**Group**  
Vertical

**Syntax**  
CH<x>:TERmination {FIFty|MEG|<NR3>}
CH<x>:TERmination?

**Arguments**  
FIFty sets the channel <x> input resistance to 50 Ω.
MEG sets the channel <x> input resistance to 1 MΩ.
<NR3> specifies the channel <x> input resistance numerically.

**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>:YUNits**

Sets or returns the units for the channel specified by <x>, where x is the channel number. String arguments are case insensitive and any unsupported units will generate an error.

Supported units are:
%

**Group**  
Vertical

**Syntax**  
CH<x>:YUNits <QString>
CH<x>:YUNits?

**Arguments**  
QString is a string of text surrounded by quotes, specifying the supported units.
**CLEARMenu (No Query Form)**

Cleans the current menu from the display. This command is equivalent to pressing the front panel Menu off.

**Group**  
Miscellaneous

**Syntax**  
CLEARMenu

**CLS (No Query Form)**

Cleans 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. The MAV bit indicates that information is in the output queue. The device clear (DCL) GPIB 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**  

**Examples**  
*CLS clears the oscilloscope status data structures.

**CURSor?**

Returns all of the current cursor settings.

**Group**  
Cursor
Syntax  CURSor?

Examples  CURSOR? might return the following as the current cursor settings:
:CURSOR:FUNCTION SCREEN;HBARS:POSITION1 0.0000;POSITION2 0.0000;UNITS BASE;:CURSOR:MODE INDEPENDENT;VBARS:POSITION1 -19.0006E-6;POSITION2 -18.9994E-6;UNITS SECONDS

CURSor:DDT? (Query Only)

Returns the cursor deltaY/deltaT (dY/dT) readout.

Conditions  This command requires a DPO4PWR application module.

Group  Cursor

Syntax  CURSor:DDT?

Examples  CURSOR:DDT? might return :CURSOR:DDT -166.6670 indicating that the cursor dV/dt read out is -166.6670

CURSor:FUNCtion

Sets or returns the cursor type. Cursors are attached to the selected waveform in Waveform mode and are attached to the display area in Screen mode.

Group  Cursor

Syntax  CURSor:FUNCtion {OFF|SCREEN|WAVEform}
CURSor:FUNCtion?

Arguments  OFF removes the cursors from the display but does not change the cursor type.
SCREEN specifies 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.

WAVEform specifies paired 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  CURSOR: FUNCTION WAVEFORM selects the paired cursors for measuring waveform amplitude and time.

CURSOR: FUNCTION? might return :CURSOR:FUNCTION SCREEN indicating that the screen cursors are currently selected.

CURSor:HBArs? (Query Only)

Returns the current settings for the horizontal bar cursors.

Group  Cursor

Syntax  CURSor:HBArs?

Examples  CURSOR:HBARs? might return the horizontal bar setting as :CURSOR:HBARs:POSITION1 320.0000E-03;POSITION2-320.0000E-03;UNITS BASE

CURSor:HBArs:DELTa? (Query Only)

Returns the vertical difference between the two horizontal bar cursors.

Group  Cursor

Syntax  CURSor:HBArs:DELTa?

Related Commands  CURSor:HBArs:UNIts

Returns  A floating point value with an exponent.

Examples  CURSOR:HBARs:DELTa? might return :CURSOR:HBARs:DELTa 5.0800E+00 indicating that the difference between the two cursors is 5.08.

CURSor:HBArs:POSITION<x>

Sets or returns the horizontal bar cursor position relative to ground, which is expressed in vertical units (usually volts). The cursor is specified by x, which can be 1 or 2.
Commands Listed in Alphabetical Order

Group  Cursor

Syntax  CURSor:HBArs:POSITION<x> <NR3>
        CURSor:HBArs:POSITION<x>?

Related Commands  CURSor:FUNCtion

Arguments  <NR3> specifies the cursor position relative to ground.

Examples  CURSOR:HBARS:POSITION1 25.0E-3 positions Cursor 1 of the horizontal cursors at 25 mV.
CURSOR:HBARS:POSITION2? might return :CURSOR:HBARS:POSITION2 -64.0000E-03 indicating that Cursor 2 of the horizontal bar cursors is at -64 mV.

CURSor:HBArs:UNIts

Sets or returns the units for the horizontal bar cursors.

Group  Cursor

Syntax  CURSor:HBArs:UNIts {BASE|PERcent}
        CURSor:HBArs:UNIts?

Arguments  BASE selects the vertical units for the selected waveform.
            PERcent selects ratio cursors.

Examples  CURSOR:HBARS:UNITS might return :CURSOR:HBARS:UNITS BASE indicating that the units for the horizontal bar cursors are base.

CURSor:HBArs:USE (No Query Form)

Sets the horizontal bar cursor measurement scale. This command is only applicable when ratio cursors are on.

Group  Cursor

Syntax  CURSor:HBArs:USE {CURRENT|HALFgrat}
Related Commands  CURSor:HBArs:UNIts

Arguments  Current sets the H Bar measurement scale so that 0% is the current position of the lowest H Bar cursor and 100% is the current position of the highest H Bar cursor.

HALFgrat sets H Bar measurement scale so that half the screen major divisions (five on the MSO/DPO4000) is 100%, where 0% is -2.5 divisions and 100% is +2.5 divisions from the center horizontal graticule.

Examples  CURSOR:HBARs:USE HALFGRAT sets the H Bar measurement scale so that five screen major divisions equals 100%.

CURSor:MODe

Sets or returns whether the two cursors move linked together in unison or separately. This command is equivalent to setting Linked to On or Off in the Cursor menu.

Group  Cursor

Syntax  CURSor:MODe {TRACk|INDependent}
        CURSor:MODe?

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. This mode only applies when the DISplay:FORMat is set to YT.

INDependent allows independent adjustment of the two cursors.

Examples  CURSOR:MODE TRACK specifies that the cursor positions move in unison.

CURSOR:MODE? might return :CURSOR:MODE TRACK indicating that the two cursors move in unison.

CURSor:VBArs? (Query Only)

Returns the current settings for the vertical bar cursors.

Group  Cursor
Syntax  CURSor:VBArs?

Examples  CURSOR:VBARS? might return the following vertical bar settings:

:CURSOR:VBArs:UNITS SECONDS;POSITION1 1.0000E-06;POSITION2 9.0000E-06

CURSor:VBArs:ALTERNATE<x>? (Query Only)

Returns the alternate readout for the waveform (Vbar) cursors specified by <x>. This alternate readout is in effect when the selected waveform is a bus or digital channel.

Group  Cursor

Syntax  CURSor:VBArs:ALTERNATE<x>?

Arguments  

x = 1 specifies vertical bar cursor1.

x = 2 specifies vertical bar cursor2.

CURSor:VBArs:DELTa? (Query Only)

Returns the horizontal difference between the two vertical bar cursors. The units are specified by the CURSor:VBArs:UNIts command.

This is equivalent to watching the cursor readout in the display while using the appropriate cursor mode.

Group  Cursor

Syntax  CURSor:VBArs:DELTa?

Related Commands  CURSor:VBArs:UNIts

Returns  <NR3>

Examples  CURSOR:VBARS:DELTA? might return :CURSOR:VBARS:DELTa 1.0640E+00 indicating that the time between the vertical bar cursors is 1.064 s.
**CURSor:VBAr:s:HPOS<x>? (Query Only)**

Returns the vertical value of the specified vertical bar ticks for cursor `<x>`.

**Group**  
Cursor

**Syntax**  
CURSor:VBAr:s:HPOS<x>?

**Related Commands**  
CURSor:VBAr:s:UNIts

**Arguments**  
<x> specifies the cursor. Valid values are 1 and 2.

**Returns**  
<NR3> indicates the value of one of the ticks. The units are the same as the selected waveform.

**Examples**  
CURSor:VBAr:s:HPOS2? might return CURSOR:VBARS:HPOS2 100E-3, indicating the waveform value where the cursor intersects it is 0.100.

**CURSor:VBAr:s:POSITION<x>**

Sets or returns the horizontal position for the specified vertical bar cursor. The cursor is specified by `<x>`, which can be 1 or 2. Values are with respect to trigger position or the zero reference point for the designated waveform (if horizontal units are not set to time). Use the CURSor:VBAr:s:UNIts command to specify units.

**Group**  
Cursor

**Syntax**  
CURSor:VBAr:s:POSITION<x> <NR3>  
CURSor:VBAr:s:POSITION<x>?

**Related Commands**  
CURSor:VBAr:s:UNIts

**Arguments**  
<NR3> specifies the cursor position.

**Returns**  
A floating point value with an exponent.
Examples

CURSor:VBArS:POSITION2 9.00E-6 positions the cursor2 vertical bar cursor at 9 ms.

CURSor:VBArS:POSITION1? this command might return :CURSOR:VBARS:POSITION1 1.0000E-06 indicating that the cursor1 vertical bar is positioned at 1 μs.

CURSor:VBArS:UNIts

Sets or returns the units for the vertical bar cursors.

Group

Cursor

Syntax

CURSor:VBArS:UNIts {SEConds|HERtz|DEGrees|PERcent}
CURSor:VBArS:UNIts?

Arguments

SEConds sets the units of the vertical bar cursors for the time domain (seconds).
HERtz sets the units of the vertical bar cursors for the frequency domain (Hertz).
DEGrees sets the units to degrees for measuring phase.
PERcent sets the units to percent for use with ratio cursors.

Returns

SECONDS, HERTZ, DEGREES, or PERCENT, depending on the current vertical bar cursor units.

Examples

CURSOR:VBARS:UNITS Hertz sets the units of the VBArs cursors to 1/seconds.
CURSOR:VBARS:UNITS? might return :CURSOR:VBARS:UNITS SECONDS indicating that the units for the vertical bar cursor are currently set to seconds.

CURSor:VBArS:USE (No Query Form)

Sets the vertical bar cursor measurement scale.

Conditions

This command is only applicable when ratio cursors are on.

Group

Cursor

Syntax

CURSor:VBArS:USE {CURRENT|HALFgrat|FIVEdivs}
Related Commands  CURSor:VBAr:s:UNIts

Arguments  CURRENT sets the V Bar measurement scale so that 0% is the current position of the left-most V Bar cursor and 100% is the current position of the right-most V Bar cursor.

HALFgrat resets the ratio range to the default positions on the screen, half of the number of horizontal divisions (five on the MSO/DPO4000), from 25% to 75% of the screen.

FIVEdivs sets V Bar measurement scale so that 5 screen major divisions is 100%, where 0% is -2.5 divisions and 100% is +2.5 divisions from the center vertical graticule.

Examples  CURSOR:VBARS:USE CURRENT sets the V Bar measurement scale to use the current cursor positions as 0% and 100% of scale if units are set to %.

CURSor:VBAr:s:VEDELTa? (Query Only)

Returns the vertical difference between the two vertical bar cursor ticks.

Group  Cursor

Syntax  CURSor:VBAr:s:VEDELTa?

Related Commands  CURSor:HBArs:UNIts

Returns  <NR3> indicates the vertical difference between the two vertical bar cursors' ticks.

Examples  CURSOR:VBARS:VDELTA? might return CURSOR:VBARS:VDELTA 1.064E+0, indicating that the vertical difference between the vertical bar cursors ticks is 1.064 units.

CURSor:XY:POLar:RADIUS:DELta? (Query Only)

Returns the difference between the cursors X radius and the cursor Y radius (ΔY, ΔX). The ratio is calculated as (cursor 2 Y - cursor 1 Y) ÷ (cursor 2 X - cursor 1 X).

Group  Cursor
Syntax CURSor:XY:POLar:RADIUS:DELta?

CURSor:XY:POLar:RADIUS:POSITION<x>? (Query Only)

Returns the polar radius for the specified cursor, where x can be either 1 or 2.

Group Cursor

Syntax CURSor:XY:POLar:RADIUS:POSITION<x>?

CURSor:XY:POLar:RADIUS:UNIts? (Query Only)

Returns the polar radius units.

Group Cursor

Syntax CURSor:XY:POLar:RADIUS:UNIts?

CURSor:XY:POLar:THETA:DELta? (Query Only)

Returns the XY cursor polar angle delta.

Group Cursor

Syntax CURSor:XY:POLar:THETA:DELta?

CURSor:XY:POLar:THETA:POSITION<x>? (Query Only)

Returns the cursor X or cursor Y polar coordinate, where x is either 1 or 2.

Group Cursor

Syntax CURSor:XY:POLar:THETA:POSITION<x>?
**CURSor:XY:POLar:THETA:UNIts? (Query Only)**

Returns the cursor coordinate units.

**Group**  
Cursor

**Syntax**  
CURSor:XY:POLar:THETA:UNIts?

**CURSor:XY:PRODUCT:DELta? (Query Only)**

Returns the difference between the cursors X position and cursor Y position. The \( \Delta X \times \Delta Y \) value is calculated as \( (X_2 - X_1) \times (Y_2 - Y_1) \).

**Group**  
Cursor

**Syntax**  
CURSor:XY:PRODUCT:DELta?

**CURSor:XY:PRODUCT:POSITION<x>? (Query Only)**

Returns the position of the X or Y cursor used to calculate the X \( \times \) Y cursor measurement, Position 1 = \( (X_1 \times Y_1) \); Position 2 = \( (X_2 \times Y_2) \). The cursor is specified by \( x \), which can be 1 or 2.

**Group**  
Cursor

**Syntax**  
CURSor:XY:PRODUCT:POSITION<x>?

**CURSor:XY:PRODUCT:UNIts? (Query Only)**

Returns the XY cursor product units.

**Group**  
Cursor

**Syntax**  
CURSor:XY:PRODUCT:UNIts?
CURSor:XY:RATIO:DELta? (Query Only)

Returns the ratio of the difference between the cursors X position and cursor Y position (ΔY, ΔX). The ratio is calculated as (Y2 - Y1) / (X2 - X1).

**Group**  
Cursor

**Syntax**  
CURSor:XY:RATIO:DELta?

CURSor:XY:RATIO:POSITION<x>? (Query Only)

Returns the X (horizontal) or Y (vertical) position for the specified cursor, which can be 1 (X) or 2 (Y). The ratio is calculated as Position 1 = (Y1/X1); Position 2 = (Y2/X2).

**Group**  
Cursor

**Syntax**  
CURSor:XY:RATIO:POSITION<x>?

CURSor:XY:RATIO:UNIts? (Query Only)

Returns the cursor X and cursor Y units for the ratio measurement.

**Group**  
Cursor

**Syntax**  
CURSor:XY:RATIO:UNIts?

CURSor:XY:READOUT

Sets or returns the XY cursor readout selection. This command is useful for oscilloscope models with limited display room. Oscilloscopes with sufficient display, such as the MSO4000 and DPO4000 series, show all the readouts simultaneously.

**Group**  
Cursor

**Syntax**  
CURSor:XY:READOUT {RECTangular|POLARcord|PRODUCT|RATio}  
CURSor:XY:READOUT?
Arguments

RECTangular specifies the XY readout as rectangular coordinates.
POLARcord specifies the XY readout as polar coordinates.
PRODUCT specifies the XY readout in X*Y format.
RATIO specifies the XY readout in X:Y format.

CURSor:XY:RECTangular:X:DELta? (Query Only)

Returns the cursor X delta value in rectangular coordinates.

Group: Cursor

Syntax: CURSor:XY:RECTangular:X:DELta?

CURSor:XY:RECTangular:X:POSITION<x>

Sets or returns the X rectangular coordinate for cursor 1 or cursor 2. The cursor is specified by x and can be either 1 or 2.

Group: Cursor

Syntax: CURSor:XY:RECTangular:X:POSITION<x> <NR3>
CURSor:XY:RECTangular:X:POSITION<x>?

Arguments: <NR3> is the coordinate in volts.

CURSor:XY:RECTangular:X:UNIts? (Query Only)

Returns the cursor X rectangular units.

Group: Cursor

Syntax: CURSor:XY:RECTangular:X:UNIts?

CURSor:XY:RECTangular:Y:DELta? (Query Only)

Returns The cursor Y delta value in rectangular coordinates.
**Group**  
Cursor

**Syntax**  
CURSor:XY:RECTangular:Y:DELta?

**CURSor:XY:RECTangular:Y:POSITION<x>**

Sets or returns the Y rectangular coordinate for cursor 1 or cursor 2. The cursor is specified by x and can be either 1 or 2.

**Group**  
Cursor

**Syntax**  
CURSor:XY:RECTangular:Y:POSITION<x> <NR3>  
CURSor:XY:RECTangular:Y:POSITION<x>?

**Arguments**  
<NR3> is the coordinate in volts.

**CURSor:XY:RECTangular:Y:UNIts? (Query Only)**

Returns the cursor Y rectangular units.

**Group**  
Cursor

**Syntax**  
CURSor:XY:RECTangular:Y:UNIts?

**CURVe**

Transfers waveform data to and from the oscilloscope in binary or ASCII format. Each waveform transferred includes a waveform preamble which contains the data format, scale, and associated information.

For analog waveforms, the CURVe? query transfers data from the oscilloscope. 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 oscilloscope returns data from the last acquisition if the source is a channel waveform that is being previewed. The data does not reflect the acquisition preview parameters. The user should always follow acquisition parameter changes with a single sequence OPC command prior to CURVe? to ensure the return data reflects the new acquisition parameters.
The CURVe command transfers waveform data to the oscilloscope. The data is stored in the reference memory location specified by \texttt{DATa:DESTination}, starting with the data point specified by \texttt{DATa:STARt}. Only one waveform can be transferred at a time. The waveform will only be displayed if the reference is displayed.

\textbf{NOTE.} Transferring large volumes of data to or from the oscilloscope takes time. ASCII waveform transfer is very inefficient.

**Group**  
Waveform Transfer

**Syntax**  
\texttt{CURVe \{<Block>|<asc curve>|DIGtal\}}\texttt{\texttt{CURVe?}}

**Related Commands**  
\texttt{DATa:DESTination, DATa:SOUrce, DATa:STARt, DATa:STOP, WFMInpre?, WFMInpre:BYT_Nr, WMOutpre?, HEADer}

**Arguments**  
\texttt{<Block>} is the waveform data in binary format. The waveform is formatted as: 
\texttt{\#<x><yyy><data><newline>}, where:

\begin{itemize}
  \item \texttt{<x>} is the number of \texttt{y} bytes. For example, if \texttt{<yyy>=500}, then \texttt{<x>=3)}
  \item \texttt{<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-bytes wide. Use the \texttt{WFMinpre:BYT_Nr} command to set the width for waveforms transferred into the oscilloscope. Use \texttt{WMOutpre:BYT_Nr} to set the width for waveforms transferred out from the oscilloscope.
  \item \texttt{<data>} is the curve data.
  \item \texttt{<newline>} is a single byte new line character at the end of the data.
\end{itemize}

\texttt{<asc curve>} is the waveform data in ASCII format. The format for ASCII data is \texttt{<NR1>[,<NR1>...]}, where each \texttt{<NR1>} represents a data point.

\texttt{DIGtal} sends the data to the specified reference waveform slot, as specified by the \texttt{DATA:DESTINATION} command. If the data encoding is a binary format, then depending upon the setting of HEADER, the binary block header is prepended to the output data. (MSO models only)

**Examples**  
\texttt{CURVe?} with ASCII encoding, start and stop of 1 and 10 respectively, and a width set to 1 might return:\texttt{CURVe 61,62,61,60,60,-59,-59,-58,-58,-59}
NOTE. Curve data is transferred from the oscilloscope asynchronously, depending on the length of the curve record. Such transfers may require several seconds to complete. During this period, the oscilloscope will not respond to the user controls. You can interrupt these asynchronous data transfers by sending a device clear message to the oscilloscope or by interrupting the query with another command or query. In order to verify that curve data has been completely transferred, it is recommended that you follow such queries with an *ESR? query to verify there are no error bits set. You can also check the event queue to determine the cause of the error. If the error was caused by an interrupted query, then the asynchronous data transfer was not complete when the *ESR? query was sent. In such cases, it may be necessary to increase the program’s time-out value to ensure that all data is transferred and read.

**DATa**

Sets or returns the format and location of waveform data transferred with the CURVe? query or CURVe command.

**Group**

Waveform Transfer

**Syntax**

DATa {INIT|SNAp}

DATa?

**Related Commands**

CURVe, DATa:STARt, DATa:STOP, DATa:ENCdg, WFMInpre:NR_Pt, WFMOutpre: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.

**Examples**

DATA? might return :DATA:DESTINATION REF1:ENCdg RIBINARY;SOURCE CH1;START 1;STOP 500;WIDTH 1

DATA INIT initializes the waveform data parameters to their factory defaults.

**DATa:DESTination**

Sets or returns the reference memory location for storing waveform data transferred into the oscilloscope by the CURVe command.
Group  Waveform Transfer

Syntax  DATa:DESTination REF<x>
        DATa:DESTination?

Related Commands  CURVe

Arguments  REF<x> is the reference location where the waveform will be stored.

Examples  DATA:DESTINATION? might return :DATA:DESTINATION REF3 indicating that reference 3 is the currently selected reference memory location for incoming waveform data. DATA:DESTINATION REF1 indicates that incoming waveform data be stored in reference 1.

**DATa:ENCdg**

Sets or returns the format of outgoing waveform data. This command is equivalent to setting WFMOutpre:ENCdg, WFMOutpre:BN_Fmt, and WFMOutpre:BYT_Or. Setting the DATa:ENGdg value causes the corresponding WFMOutpre values to be updated and conversely.

**NOTE. This command and query does not apply to incoming waveform data.**

Group  Waveform Transfer

Syntax  DATa:ENCdg
        {ASCIi|FAStest|RIBinary|RPBinary|SRIBinary|SRPbinary}
        DATa:ENCdg?

Related Commands  WFMOutpre:ENCdg, WFMOutpre:BN_Fmt, WFMOutpre:BYT_Or

Arguments  ASCIi specifies the ASCII representation for waveform data points. If ASCII is the value, then :BN_Fmt and :BYT_Or are ignored.

FAStest specifies that the data be sent in the fastest possible manner consistent with maintaining accuracy and is interpreted with respect to the waveform specified by DATa:SOUrce.

RIBINARY specifies signed integer data point representation with the most significant byte transferred first.
When \( :\text{BYT}_\text{Nr} \) is 1, the range is from -128 through 127. When \( :\text{BYT}_\text{Nr} \) is 2, the range is from -32,768 through 32,767. 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 \( :\text{BYT}_\text{Nr} \) is 1, the range from 0 through 255. When \( :\text{BYT}_\text{Nr} \) is 2, the range is from 0 to 65,535. 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.

**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.

### Table 2-39: DATa and WFMOutpre Parameter Settings

<table>
<thead>
<tr>
<th>DATa:ENCdg Setting</th>
<th>WFMOutpre Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>:ENCdg</td>
</tr>
<tr>
<td>ASCii</td>
<td>ASC</td>
</tr>
<tr>
<td>FASTest</td>
<td>BIN</td>
</tr>
<tr>
<td>RIBinary</td>
<td>BIN</td>
</tr>
<tr>
<td>RPBinary</td>
<td>BIN</td>
</tr>
<tr>
<td>SRIBinary</td>
<td>BIN</td>
</tr>
<tr>
<td>SRPBinary</td>
<td>BIN</td>
</tr>
</tbody>
</table>

#### Examples

**DATA:ENCdg?** might return **:DATa:ENCdg SRPBINARY** for the format of the outgoing waveform data.

**DATA:ENCdg RPBinary** sets the data encoding format to be a positive integer where the most significant byte is transferred first.

### DATa:SOURce

Sets or returns the location of the waveform data transferred from the oscilloscope by the **CURVe?** query.

#### Group

Waveform Transfer
Syntax

```
DATa:SOURce
{CH1|CH2|CH3|CH4|MATH|REF1|REF2|REF3|REF4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9}
```

```
DATa:SOURce?
```

**Related Commands**

```
CURVe
```

**Arguments**

CH1–CH4 specifies which analog channel data will be transferred from the oscilloscope to the controller, channels 1 through 4.

MATH specifies that the Math waveform data will be transferred from the oscilloscope to the controller.

REF1–REF4 specifies which Reference waveform data will be transferred from the oscilloscope to the controller, waveforms, 1 through 4.

D0–D15 specifies which digital channel data will be transferred from the oscilloscope to the controller, channels 0 through 15. (MSO models only)

DIGital specifies that the Digital waveform data will be transferred from the oscilloscope to the controller. (MSO models only)

**Examples**

```
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 CH1
```

specifies that the CH1 waveform will be transferred in the next CURVe? query.

**DATa:STARt**

Sets or returns the starting data point for incoming or outgoing waveform transfer. This command allows for the transfer of partial waveforms to and from the oscilloscope.

**Group**

Waveform Transfer

**Syntax**

```
DATa:STARt <NR1>
```

```
DATa:STARt?
```

**Related Commands**

```
CURVe, DATa, DATa:STOP, WFMInpre:NR_Pt, 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:STOP? might return :DATA:START 214 indicating that data point 214 is the first waveform data point that will be transferred.

DATa:START 10 specifies that the waveform transfer will begin with data point 10.

**DATa:STOP**

Sets or returns the last data point that will be transferred when using the CURVe? query. This command allows for the transfer of partial waveforms from the oscilloscope.

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.

**Group**

Waveform Transfer

**Syntax**

DATa:STOP <NR1>

DATa:STOP?

**Related Commands**

CURVe, DATa, DATa:STARt, WFMInpre:NR_Pt, 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? might return :DATA:STOP 14900 indicating that 14900 is the last waveform data point that will be transferred.
DATA:STOP 15000 specifies that the waveform transfer will stop at data point 15000.

**DATE**

Sets or returns the date the oscilloscope displays.

**Group**  
Miscellaneous

**Syntax**  
DATE <QString>
DATE?

**Related Commands**  
TIME

**Arguments**  
<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**  
DATE "2006-01-24" specifies that the date is set to January 24, 2006.
DATE? might return :DATE 2006-01-24 indicating the current date is set to January 24, 2006.

**/DDT**

Allows you to specify a command or a list of commands that execute when the oscilloscope receives a *TRG command or the GET IEEE488.2 interface message. 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**

*DDT #OACQUIRE:STATE RUN specifies that the acquisition system will be started each time a *TRG command is sent.

**DESE**

Sets or returns 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 detailed discussion of the use of these registers, see Registers.

**Group** Status and Error

**Syntax**

```
DESE <NR1>
DESE?
```

**Related Commands**


**Arguments**

<NR1> sets the binary bits of the DESER 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.

**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.

**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.
**DESkew (No Query Form)**

Causes the deskew values for all channels to be set to the recommended values. Equivalent to pressing the "Set all deskews to recommended values" button in the application UI.

**NOTE.** The actual deskew values for each channel can be set or queried using the `CH<x>:DESkew` command.

**Group**

Vertical

**Syntax**

DESkew {SETALLtorec}

**Arguments**

<SETALLtorec> sets the deskew for all channels to the recommended values.

**DESkew:DISPlay**

Sets or returns the state of the deskew table display.

**Group**

Vertical

**Syntax**

DESkew:DISPlay {OFF|ON|0|1}

DESkew:DISPlay?

**Arguments**

OFF or 0 turns off the deskew table display.

ON or 1 turns on the deskew table display.

**DIAg:LOOP:OPTion**

Sets the self-test loop option.

**Group**

Calibration and Diagnostic

**Syntax**

DIAg:LOOP:OPTion {ALWAYS|FAIL|ONFAIL|ONCE|NTIMES}

**Arguments**

ALWAYS continues looping until the self tests (diagnostics) are stopped via the front panel or by an oscilloscope command.
FAIL causes looping until the first self test (diagnostic) failure or until self tests (diagnostics) are stopped.

ONFAIL causes looping on a specific test group as long as a FAIL status is returned from the test.

ONCE executes self test (diagnostics test) sequence once.

NTIMES runs “n” number of loops.

**Examples**

```
DIAG:LOOP:OPTION ONCE runs one loop of self tests.
```

**DIAG:LOOP:OPTION:NTIMES**

Sets the self-test loop option to run N times.

**Group**

Calibration and Diagnostic

**Syntax**

```
DIAG:LOOP:OPTION:NTIMES <NR1>
DIAG:LOOP:OPTION:NTIMES?
```

**Arguments**

<NR1> is the number of self-test loops.

**Examples**

```
DIAG:LOOP:OPTION:NTIMES 3 sets the self-test loop to run three times.

```

**DIAG:LOOP:STOP (No Query Form)**

Stops the self-test at the end of the current loop.

**Group**

Calibration and Diagnostic

**Syntax**

```
DIAG:LOOP:STOP
```

**Examples**

```
DIAG:LOOP:STOP stops the self test at the end of the current loop.
```
DIAg:RESULT:FLAg? (Query Only)

Returns the pass/fail status from the last self-test sequence execution. Use this query to determine which test(s) has failed.

Group Calibration and Diagnostic

Syntax DIAg:RESULT:FLAg?

Related Commands DIAg:RESULT:LOG?

Returns PASS indicates that all of the selected self (diagnostic) tests have passed.
FAIL indicates that at least one of the selected self (diagnostic) tests has failed.
IN PROGRESS indicates that at least one of the selected self (diagnostic) tests is ongoing.
NOT RUN indicates that the selected diagnostic test suite has not been run since the instrument was powered on.

Examples DIAg:RESULT:FLAG? might return DIAg:RESULT:FLAG PASS.

DIAg:RESULT:LOG? (Query Only)

Returns the internal results log from the last self-test sequence execution. The list contains all modules and module interfaces that were tested along with the pass/fail status of each.

Group Calibration and Diagnostic

Syntax DIAg:RESULT:LOG?

Related Commands DIAg:RESULT:FLAg?

Returns <QString> in the following format:
<Status>--<Module name>[,<Status>--<Module name>...]
Examples

DIAG:RESULT:LOG? might return:

```
DIAG:RESULT:LOG "NOT RUN--CPU, NOT RUN--DISPLAY, NOT RUN--FPANEL, NOT RUN--IO, NOT RUN--ACQ, NOT RUN--ROM, NOT RUN--APPKEY"
```

**DIAg:SELect (No Query Form)**

Sets the type of diagnostics grouping.

**Group**
Calibration and Diagnostic

**Syntax**

```
DIAg:SELect {ALL|APPKey|CPU|DISplay|FPAnel|IO|ROM|ACQ}
```

**Arguments**

- ALL runs all diagnostic groups.
- APPKey runs just the application key diagnostic group.
- CPU runs just the CPU diagnostic group.
- DISPLAY runs just the display circuit diagnostic group.
- FFPANEL runs just the front panel diagnostic group.
- IO runs just the IO board diagnostic group.
- ROM runs just the IO board diagnostic group.
- ACQ runs just the acquisition system diagnostic group.

**DIAg:SELect:<function> (No Query Form)**

Runs self-tests on the specified system subsystem.

**Group**
Calibration and Diagnostic

**Syntax**

```
DIAg:SELect:<function>
```

**Arguments**

<function> specifies a single oscilloscope subsystem on which to run self tests (diagnostics). Valid values are:

- ACQ tests the acquisition system.
- APPKey tests the application keys.
- CPU tests the CPU.
- DISPLAY tests the display.
FPAnel tests the front panel controls.
IO tests the IO ports.
ROM tests the system read only memory.

**Examples**

```
DIAG:SELECT:CPU
```
sets the oscilloscope to run just CPU tests.

### DIAg:STATE (No Query Form)

This command starts or stops the oscilloscope self-test. Depending on the argument, self-test capabilities are either turned on or off.

**Group** Calibration and Diagnostic

**Syntax**

```
DIAG:STATE {EXECute|ABORt}
```

**Arguments**

EXECute starts diagnostics.

ABORt stops diagnostics at the end of the current loop.

**Examples**

```
DIAG:STATE EXECute
```
starts diagnostics.

### DISplay? (Query Only)

Returns the current display settings.

**Group** Display

**Syntax**

```
DISPLAY?
```

### DISplay:CLOCk

Sets or returns whether the oscilloscope displays the date and time. The query form of this command returns an ON (1) or an OFF (0).

**Group** Display
Syntax

\texttt{DISplay:CLOCK \{ON|OFF|<NR1>\}}
\texttt{DISplay:CLOCK?}

Related Commands

DATE, TIME

Arguments

ON enables the display of date and time.
OFF disables the display of date and time.
\(<\text{NR1}> = 0\) disables the display of date and time; any other value enables the display of date and time.

Examples

\texttt{DISPLAY:CLOCK ON} enables display of date and time.
\texttt{DISPLAY:CLOCK?} might return \texttt{:DISPLAY:CLOCK 1} indicating that the display of date and time is currently enabled.

\textbf{DISplay:DIGital:HEIght}

Sets or returns the number of available digital waveform position slots.

\begin{quote}
\textbf{NOTE.} If the number of live digital channels exceeds the number of slots, their height is reduced to a smaller size, allowing all digital channels to be displayed.
\end{quote}

Group

Display

Syntax

\texttt{DISplay:DIGital:HEIght \{SMAll|MEDium|LARge\}}
\texttt{DISplay:DIGital:HEIght?}

Arguments

SMAll sets the height to 40.
MEDium sets the height to 20.
LARge sets the height to 10.

\textbf{DISplay:FORMat}

Sets or returns the display format.

Group

Display
**Syntax**

```
DISPLAY:FORMat {YT|XY}
DISPLAY:FORMat?
```

**Arguments**

YT sets the display to a voltage versus time format and is the default mode.

XY argument displays one waveform against another. Selecting one source causes its corresponding source to be implicitly selected, producing a single trace from the two input waveforms.

**Examples**

```
DISPLAY:FORMat XY sets the display format to XY.
DISPLAY:FORMat? might return DISPLAY:FORMat YT indicating the display format is YT.
```

**DISplay:GRAticule**

Selects or queries the type of graticule the oscilloscope displays.

**Group**

Display

**Syntax**

```
DISPLAY:GRAticule {CROSSHair|FRAme|FULl|GRID}
DISPLAY:GRAticule?
```

**Arguments**

CROSSHair specifies a frame and cross hairs.

FRAme specifies a frame only.

FULl specifies a frame, a grid and cross hairs.

GRID specifies a frame and grid only.

**Examples**

```
DISPLAY:GRAticule FRAme sets the graticule type to display the frame only.
DISPLAY:GRAticule? might return DISPLAY:GRAticule FULL indicating that all graticule elements are selected.
```

**DISplay:INTENSITy?** *(Query Only)*

Returns the display intensity settings.

**Group**

Display
Syntax

DISplay:INTENSITY?

Examples

DISPLAY:INTENSITY? might return: :DISPLAY:INTENSITY:WAVEFORM 30;GRATICULE 75;BACKLIGHT HIGH

DISplay:INTENSITY:BACKLight

Sets and returns the waveform backlight intensity settings.

Group

Display

Syntax

DISplay:INTENSITY:BACKLight {LOW|MEdium|HIGH}

Examples

DISPLAY:INTENSITY:BACKLight? might return
DISPLAY:INTENSITY:BACKLIGHT HIGH

DISplay:INTENSITY:GRAticule

Sets and returns the display graticule intensity settings.

Group

Display

Syntax

DISplay:INTENSITY:GRAticule <NR1>

Arguments

<NR1> is the graticule intensity and ranges from 0 to 100 percent.

Examples

DISPLAY:INTENSITY:GRATICULE? might return
DISPLAY:INTENSITY:GRATICULE 30

DISplay:INTENSITY:WAVEform

Sets and returns the display waveform intensity settings.

Group

Display
Commands Listed in Alphabetical Order

**Syntax**

```plaintext
DISPLAY:INTENSITY:WAVEform <NR1>
DISPLAY:INTENSITY:WAVEform?
```

**Arguments**

<NR1> is the waveform intensity and ranges from 1 to 100 percent.

**Examples**

```plaintext
DISPLAY:INTENSITY:WAVEform? might return
DISPLAY:INTENSITY:WAVEform 60
```

as the intensity of the waveforms.

**DISplay:PERSistence**

Sets or returns the display persistence. This affects the display only.

**NOTE.** When Persistence is set to Infinite, it does not mean that the brightness of any pixel should never decrease. The brightness of a pixel is proportionally dependent on the ratio between its intensity (which does NOT decrease at Infinite Persistence) and the maximum value of intensity of any pixel on the screen. If a particular pixel get hit less often than others, its brightness will decrease over time. It will become less bright relative to the pixels that get hit often.

**Group**

Display

**Syntax**

```plaintext
DISPLAY:PERSistence {<NR3>|CLEAR|AUTO|MINimum|INFInite}
DISPLAY:PERSistence?
```

**Arguments**

<NR3> specifies the time of the persistence.

CLEAR resets the persist time count down and clears the display of acquired points.

INFInite displays waveform points until a control change resets the acquisition system. When persistence is set to infinite, it does not mean that the brightness of any pixel should never decrease. The brightness of a pixel is proportionally dependent on the ratio between its intensity, which does NOT decrease at infinite persistence, and the maximum value of intensity of any pixel on the screen. Thus, if a particular pixel gets hit less often than others, its brightness will decrease over time. It will become less bright relative to the pixels that get hit often.

AUTO specifies that the oscilloscope automatically determines the best waveform persistence based on the value of waveform intensity (DISPLAY:INTEnsITY:WAVEFORM)

MINimum specifies that the waveform persistence is set to the minimum value of 0.0E0.
Examples

`DISPLAY:PERSISTENCE 3` specifies that the waveform points are displayed fading for 3 seconds before they completely disappear.

**DISplay:STYle:DOTsonly**

Turns on or off the dots-only mode for the waveform display.

**Group**

Display

**Syntax**

`DISPLAY:STYLE:DOTsonly {ON|OFF|<NR1>}`

`DISPLAY:STYLE:DOTsonly?`

**Arguments**

- ON or `<NR1> ≠ 0` turns on the dots-only display.
- OFF or `<NR1> = 0` turns off the dots-only display.

**D<x>**

Sets or returns parameters for digital channel `<x>`, where x is the channel number.

**Group**

Vertical

**Syntax**

`D<x>`

**D<x>:LABel**

Sets or returns the waveform label for digital channel `<x>`, where x is the channel number.

**Group**

Vertical

**Syntax**

`D<x>:LABel <Qstring>`

`D<x>:LABel?`

**Arguments**

- `<Qstring>` is an alphanumeric string of text, enclosed in quotes, that contains the text label information for the digital channel `<x>` waveform. The text string is limited to 30 characters.
Examples

D4:LABEL? might return :D4:LABEL “Clk Wfm 4”.

D<x>:POSITION

Sets or returns the vertical position for digital channel <x>, where x is the channel number.

**NOTE.** Unlike analog channels, which use the center baseline as the zero point, digital channels use the bottom logic level as the zero point. Because of the slot model used to describe digital channel position, the oscilloscope rounds off divisions to the nearest slot.

Group
Vertical

Syntax
D<x>:POSITION <NR3>
D<x>:POSITION?

Arguments
<NR3> is the position of the digital channel in slot units. Use the oscilloscope front-panel controls to place the channel; then query the channel to obtain an exact value for the position.

D<x>:THRESHOLD

Sets or returns the logical threshold for digital channel <x>, where x is the channel number.

Group
Vertical

Syntax
D<x>:THRESHOLD {ECL|TTL|<NR3>}
D<x>:THRESHOLD?

Arguments
ECL sets the digital threshold for channel <x> to a preset ECL high level of –1.3V.
TTL sets the digital threshold for channel <x> to a preset TTL high level of 1.4V.
<NR3> specifies the digital threshold for channel <x> in volts.

Examples
D5:THRESHOLD ECL sets the threshold for D5 digital channel to ECL levels.
D14:THRESHOLD? might return :D14:THRESHOLD 1.2500 indicating that the threshold level for the D14 digital channel is 1.25 volts.
**ESE**

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 detailed discussion on how to use 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.

**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 the binary value 10111010.

**ESR? (Query Only)**

Returns the contents of the Standard Event Status Register (SESR). *ESR? also clears the SESR (reading the SESR clears it). For a detailed discussion on how to use registers, see Registers.

**Group**  Status and Error

**Syntax**  

*ESR?

**Related Commands**  

ALLEv?, *CLS, DESE, *ESE, EVENT?, EVMsg?, *SRE, *STB?
**Examples**  *ESR?* might return 213, showing that the SESR contains the binary value 11010101.

**ETHERnet:DHCPbootp**

Sets or returns the network initialization search for a DHCP/BOOTP server.

**Group**  Ethernet

**Syntax**  
ETERnet:DHCPbootp {ON|OFF}
ETHERnet:DHCPbootp?

**Arguments**  
ON enables the oscilloscope to search the network for a DHCP or BOOTP server in order to automatically assign a dynamic IP address to the oscilloscope.

**NOTE.** Do not use DHCP/BOOTP searching if your oscilloscope has been assigned a static address on a network. If you set this command to ON, the DHCP/BOOTP search will delete or change your static IP address information.

OFF disables the oscilloscope to search the network for a DHCP or BOOTP server.

**Examples**  
ETHERNET:DHCPBOOTP ON sets the oscilloscope to search for a DHCP or BOOTP server and assign a dynamic IP address to the oscilloscope.

**ETHERnet:DNS:IPADDress**

Sets or returns the network Domain Name Server (Dns) IP address.

**Group**  Ethernet

**Syntax**  
ETERnet:DNS:IPADDress <QString>
ETHERnet:DNS:IPADDress?

**Arguments**  
<QString> is a standard IP address value, enclosed in quotes.

**Examples**  
ETHERNET:DNS:IPADDRESS “128.196.13.352” sets the Dns IP address that the oscilloscope uses to communicate with the network.
**ETHERnet:DOMAINname**

Sets or returns 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 value assigned to the oscilloscope. This is assigned at the factory and cannot 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**

Sets or returns the remote interface gateway IP address.

**Group** Ethernet

**Syntax**

ETHERnet:GATEWay:IPADDress <Qstring>
ETHERnet:GATEWay:IPADDress?

**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:HTTPPort**

Sets or returns the remote interface HTTP port value.

**Group**

Ethernet

**Syntax**

ETHERnet:HTTPPort <QString>
ETHERnet:HTTPPort?

**Arguments**

<QString> is an integer port number, enclosed in quotes.

---

**NOTE.** Consider the following if you are using the e*Scope™ control software. If you don’t enter a port address in the URL, then the ETHERnet:HTTPPort value must be set to "80", which is the default port for HTTP protocol. If you use a URL with a port address (for example: http://DPO4104-04WKL4:1234), the port number is specified by the number after the colon. Set the ETHERnet:HTTPPort value to this same number.

---

**Examples**

ETHERNET:HTTPPORT “80” sets the HTTP port value to 80.

---

**ETHERnet:IPADDress**

Sets or returns the IP address assigned to the oscilloscope.

**Group**

Ethernet

**Syntax**

ETHERnet:IPADDress <QString>
ETHERnet: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:NAME**

Sets or returns the network name assigned to the oscilloscope.

**Group**  
Ethernet

**Syntax**  
ETHERnet:NAME <QString>  
ETHERnet:NAME?

**Arguments**  
<doubleString> is the network name assigned to the oscilloscope, enclosed in quotes.

**Examples**  
ETHERNET:NAME “labscope1” sets the oscilloscope’s network name.

**ETHERnet:PASSWORD**

Sets or returns the HTTP Ethernet access password. If a password is set, the user must enter the password before the user’s Web browser can access the oscilloscope.

**Group**  
Ethernet

**Syntax**  
ETHERnet:PASSWORD <new>  
ETHERnet:PASSWORD?

**Arguments**  
<new> is a new password, enclosed in quotes.

**Examples**  
ETHERNET:PASSWORD "ZEN53" replaces the current Ethernet password with the new password ZEN53.  

**ETHERnet:PING (No Query Form)**

Causes the oscilloscope to ping the gateway IP address.

**Group**  
Ethernet

**Syntax**  
ETHERnet:PING EXECute
ETHERnet:PING:STATUS? (Query Only)

Returns the results from sending the ETHERnet:PING command to ping the gateway IP address.

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.
INPROGRESS is returned if the ping operation is still executing.

ETHERnet:SUBNETMask

Sets or returns the remote interface subnet mask value.

Group Ethernet

Syntax ETHERnet:SUBNETMask <QString>
ETHERnet:SUBNETMask?

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)

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. (See page 3-12, Messages.)

EVMsg? (Query Only)

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 along with an explanatory message. For 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:

<Event Code><Comma><QString>[<Event Code><Comma><QString>...]<QString>::= <Message>[<Command>] where <Command> is the command that caused the error and may be returned when a command error is detected by the oscilloscope. 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)

Returns the number of event codes in the Event Queue. This is useful when using the ALLEv? query, which returns the exact number of events.
Commands Listed in Alphabetical Order

**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.

**FACTory (No Query Form)**

Resets the oscilloscope to its factory default settings.

This command does the following:
- Clears the Event Status Enable Register
- Clears the Service Request Enable Register
- Sets the Device Event Status Enable Register to 255
- Purges all defined aliases
- Enables all Command Headers
- Sets the macro defined by *DDT to a "zero-length field"
- Clears the pending operation flag and associated operations

This command does not reset the following:
- Communication settings
- Selected GPIB address.
- State of the VXI-11 (Ethernet IEEE Std 488.2) interface.
- Calibration data that affects device specifications
- Protected user data
- Stored settings
- Power On Status Clear Flag
- Oscilloscope password

**Group** Save and Recall

**Syntax** FACTory
**Related Commands**

*PSC, *RCL, RECall:SETUp, *RST, *SAV, SAVe:SETUp

**Arguments**

None

**Examples**

FACTORY resets the oscilloscope to its factory default settings.

---

**FILESystem? (Query Only)**

Returns the directory listing of the current working directory and the number of bytes of free space available. This query is the same as the FILESystem:DIR? query and the FILESystem:FREESpace? query.

**Group**

File System

**Syntax**

FILESystem?

**Related Commands**

FILESystem:CWD, FILESystem:DELEte, FILESystem:DIR?, FILESystem:REName

**Arguments**

None.

**Examples**

FILESYSTEM? might return

:FILESYSTEM:DIR
"tek00000.bmp","elusiveGlitch1.png","TEMP.TMP",
"file1.wfm","file2.wfm", "MATH1.wfm","REF1.wfm","REF2.wfm";FREESPACE 30212096

---

**FILESystem:CWD**

Sets or returns the current working directory (CWD) for FILESystem commands. The default working directory is "D:". Anytime you use this command to change the directory, the directory that you specify is retained as the current working directory until you either change the directory or you delete the directory. If you delete the current working directory, the oscilloscope resets current working directory to the default directory (D:) the next time the oscilloscope is powered on or the next time you execute a file system command.
This command supports the permutations of file and directory names supported by Microsoft Windows:

- Relative path names; for example, "./Temp"
- Absolute path names; for example, "D:/MyWaveform"
- Implied relative path names; for example "newfile.txt" becomes "D:/TekScope/newfile.txt" if the current working directory is "D:/TekScope"

**Group**  
File System

**Syntax**  
FILESystem:CWD {<new working directory path>}

**Arguments**  
<new working directory path> is a quoted string that defines the current working; a directory name can be up to 128 characters.

**Examples**  
FILESYSTEM:CWD "D:/TekScope/images" sets the current working directory to images.

FILESYSTEM:CWD? might return :FILESYSTEM:CWD "D:/TekScope/Waveforms" indicating that the current working directory is set to Waveforms.

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

This command deletes a named file. If you specify a directory name, it will delete the directory and all of its contents, the same as the RMDir command. You can also specify the filename as *.* to delete all of the files in the current or specified directory.

**Group**  
File System

**Syntax**  
FILESystem:DELEte <file path>

**Related Commands**  
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.
The argument \texttt{*.} will delete all files and subdirectories within the current working directory.

**Examples**

FILESYSTEM:DELETE "NOT_MINE_SET" deletes the file named \texttt{NOT_MINE_SET} from the current working directory.

**FILESystem:DIR? (Query Only)**

Returns a list of quoted strings. Each string contains the name of a file or directory in the current working directory.

**Group**

File System

**Syntax**

FILESystem:DIR?

**Related Commands**

FILESystem:CWD, FILESystem:MKDir

**Arguments**

None

**Returns**

FILESystem:DIR? returns a list of files and directories in the current working directory.

**Examples**

FILESYSTEM:DIR? might return

:FILESYSTEM:DIR
"tek00000.png","my_CAN_setup.set","savedwfml.isf","myImages"

**FILESystem:FORMAT (No Query Form)**

Formats a mass storage device. This command should be used with extreme caution as it causes all data on the specified mass storage device to be lost. Drive letters (e.g., E:) are case sensitive and must be upper case. For all other FILESYSTEM commands, drives letters are not case sensitive. Example: FILES:FORMAT "E:\" Formats the USB flash drive installed in the oscilloscope’s front panel USB port.

**Group**

File System

**Syntax**

FILESystem:FORMAT
Arguments: <drive name> is a quoted string that defines the disk drive to format.

Examples: FILESYSTEM:FORMAT "E:"

Formats the USB flash drive installed in the oscilloscope’s front panel USB port.

FILESystem:FREESpace? (Query Only)

Returns the number of bytes of free space on the current drive.

Group: File System

Syntax: FILESystem:FREESpace?

Related Commands: FILESystem:FREESpace?, FILESystem:CWD

FILESystem:LDIR? (Query Only)

Returns a semicolon separated list of every file and directory in the folder referred to by the FILESystem:CWD command. This is different than the FILESystem: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:

<file name>;<type DIR or FILE>;<size in bytes>;<date>;<time>

Group: File System

Syntax: FILESystem:LDIR?

Related Commands: FILESystem:CWD, FILESystem:DIR?

Returns: A string of the form: <file name>;<type DIR or FILE>;<size in bytes>;<date>;<time>

**FILESystem:MKDir (No Query Form)**

Creates a new folder.

**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 "D:/NewDirectory" creates the directory named NewDirectory at the root of the D drive.

These two commands create the directory MyNewSubDirectory within the existing directory MyDirectory at the root of the D drive:

```
FILESYSTEM:CWD "D:/MyDirectory"; FILESYSTEM:MKDIR "MyNewSubDirectory"
```

This, of course, assumes that D:/MyDirectory already existed and was not a read-only directory.

**FILESystem:READFile (No Query Form)**

 Writes the contents of the specified file to the specified 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, specify only the file name.
**Examples**  
`FILESYSTEM:READFILE "D:/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)**

Assigns a new name to an existing file.

**Group**  
File System

**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 name and path. If the file path is within the current working directory, you need only specify the file name.
- `<new 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:RENAME "D:/TEK00000.SET","D:/MYSETTING.SET"` gives the file named TEK00000.SET the new name of MYSETTING.SET. The file remains in the root directory on the D drive.

**FILESystem:RMDir (No Query Form)**

Deletes a named directory. This command deletes the specified directory and all of its contents. The directory must not be a read-only directory.

**Group**  
File System

**Syntax**  
`FILESystem:RMDir <directory path>`

**Related Commands**  
`FILESystem:CWD`

**Arguments**  
- `<directory path>` is a quoted string that defines the directory name and path. If the file path is within the current working directory, you need only specify the file name.
**Examples**

```
FILESYSTEM:RMDIR "D:/OldDirectory"
```

removes the directory named OldDirectory from the root of the D drive.

---

**FILESystem:WRITEFile (No Query Form)**

Writes the specified block data to a file in the oscilloscope current working directory. If the specified file does not exist or is not readable, an appropriate error event is posted.

**Group**

File System

**Syntax**

```
FILESystem:WRITEFile <file path>, <data>
```

**Related Commands**

FILESystem:CWD

**Arguments**

- `<file path>` is the quoted string that defines the file name and path. If the path is within the current working directory, specify the file name only.
- `<data>` can be either DEFINITE LENGTH encoding or INDEFINITE LENGTH ARBITRARY BLOCK PROGRAM DATA encoding as described in IEEE488.2.

---

**FPAnel:PRESS (No Query Form)**

Simulates the action of pressing a specified front-panel button.

When the front panel is locked, the front-panel button and multipurpose knob operations are suspended. The FPAnel:PRESS and the FPAnel:TURN commands will also not work. You can work around this by using the appropriate programmatic interface commands, instead of the front-panel commands.

**Group**

Miscellaneous

**Syntax**

```
FPAnel:PRESS <button>
```

**Arguments**

- `<button>` is the name of a front-panel button. Most of the argument names associate directly with their front panel buttons. For example, AUTOSet is for the Autoset button. The few commands that do not have obvious associations are listed below.
### Table 2-40: FPAnel:PRESS arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQuire</td>
<td>Acquire button</td>
</tr>
<tr>
<td>AUTOset</td>
<td>Autoset button</td>
</tr>
<tr>
<td>BMENU&lt;x&gt;</td>
<td>Screen bottom menu buttons, where &lt;x&gt;=1 for the left-most bottom menu button and &lt;x&gt;=7 for the right-most bottom menu button</td>
</tr>
<tr>
<td>B&lt;x&gt;</td>
<td>Bus select buttons, where &lt;x&gt; = 1 through 4.</td>
</tr>
<tr>
<td>CH&lt;x&gt;</td>
<td>Channel select button, where &lt;x&gt;=1 for channel 1, &lt;x&gt;=2 for channel 2, and so on</td>
</tr>
<tr>
<td>CURsor</td>
<td>Cursors button</td>
</tr>
<tr>
<td>D15D0</td>
<td>D0–D15 button</td>
</tr>
<tr>
<td>DEFaultsetup</td>
<td>Default Setup button</td>
</tr>
<tr>
<td>Fine</td>
<td>Fine button</td>
</tr>
<tr>
<td>FORCetrig</td>
<td>Force Trig button</td>
</tr>
<tr>
<td>HARDcopy</td>
<td>Hardcopy button</td>
</tr>
<tr>
<td>INTensity</td>
<td>Intensity button</td>
</tr>
<tr>
<td>MAGnify</td>
<td>Magnify (zoom) button (not the zoom/pan knob)</td>
</tr>
<tr>
<td>MARk</td>
<td>Mark Set/Clear button</td>
</tr>
<tr>
<td>MATH</td>
<td>M button</td>
</tr>
<tr>
<td>MENUOff</td>
<td>Menu Off button</td>
</tr>
<tr>
<td>MEASurement</td>
<td>Measure button</td>
</tr>
<tr>
<td>NEXT</td>
<td>Next arrow button</td>
</tr>
<tr>
<td>PAUse</td>
<td>Play/pause button</td>
</tr>
<tr>
<td>PREv</td>
<td>Previous arrow button</td>
</tr>
<tr>
<td>REF</td>
<td>R button</td>
</tr>
<tr>
<td>RMENU&lt;x&gt;</td>
<td>Screen side menu buttons, where &lt;x&gt;=1 for the top-most side menu button and &lt;x&gt;=5 for the bottom-most side menu button</td>
</tr>
<tr>
<td>RUinstop</td>
<td>Run/Stop button</td>
</tr>
<tr>
<td>SAVEBUtton</td>
<td>Save button</td>
</tr>
<tr>
<td>SAVERecall</td>
<td>Save/Recall Menu button</td>
</tr>
<tr>
<td>SEArch</td>
<td>Search button</td>
</tr>
<tr>
<td>SELECT</td>
<td>Select button</td>
</tr>
<tr>
<td>SETTO50</td>
<td>Set to 50% button</td>
</tr>
<tr>
<td>SINGleseq</td>
<td>Single button</td>
</tr>
<tr>
<td>TEST</td>
<td>Test button</td>
</tr>
<tr>
<td>TRIGger</td>
<td>Trigger Menu button</td>
</tr>
<tr>
<td>UTILity</td>
<td>Utility button</td>
</tr>
</tbody>
</table>
Examples

FPANEL:PRESS AUTOSET executes the oscilloscope Autoset function.

**FPAnel:TURN (No Query Form)**

Simulates the action of turning a specified front-panel control knob.

When the front panel is locked, the front-panel button and multipurpose knob operations are suspended. The FPAnel:PRESS and FPAnel:TURN commands will also not work, and, they will not generate an error. You can work around this 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**

FPAnel:TURN <knob>,<n>

**Arguments**

<n> is the name of a rotating control.

A comma (,) separates the control knob argument from the numeric rotation value argument. You do not need a white space between the arguments and the comma.

<n> represents the rotation direction and magnitude of rotation. Negative values represent a counterclockwise knob rotation, and positive values represent a clockwise rotation. The magnitude of <n> specifies the amount of the turn, where <n> = 1 represents turning the knob one unit, <n> = 2 represents turning the knob two units, <n> = 4 represents turning the knob four units, and so on. The range of units depends on which front panel knob is specified.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPKNOB1</td>
<td>Multipurpose a knob</td>
</tr>
<tr>
<td>GPKNOB2</td>
<td>Multipurpose b knob</td>
</tr>
<tr>
<td>HORZPos</td>
<td>Horizontal Position knob</td>
</tr>
<tr>
<td>HORZScale</td>
<td>Horizontal Scale knob</td>
</tr>
<tr>
<td>PANKNOB1</td>
<td>Outer pan knob</td>
</tr>
<tr>
<td>TRIGLevel</td>
<td>Trigger Level knob</td>
</tr>
<tr>
<td>VERTPOS&lt;n&gt;</td>
<td>Vertical Position knob</td>
</tr>
<tr>
<td>VERTSCALE&lt;n&gt;</td>
<td>Vertical Scale knob</td>
</tr>
<tr>
<td>ZOOM</td>
<td>Inner zoom knob</td>
</tr>
</tbody>
</table>

Table 2-41: FPAnel:TURN arguments
Examples
FPANEL: TURN TRIGLEVEL, 10 duplicates turning the front-panel Trigger Level knob clockwise by 10 units.

**GPIBUsb:ADDress? (Query Only)**

Returns the current GPIB address setting for a connected TEK-USB-488 adaptor module.

**Group** Miscellaneous

**Syntax** GPIBUsb:ADDress?

**GPIBUsb:ID? (Query Only)**

Returns the identification string of the connected TEK-USB-488 adaptor module and firmware version. If a TEK-USB-488.2 module is not connected, the system returns “Not detected”.

**Group** Miscellaneous

**Syntax** GPIBUsb:ID?

**HARDCopy (No Query Form)**

Sends a hard copy of the screen display to the currently active printer using the current palette and layout settings.

**Group** Hard Copy

**Syntax** HARDCopy {START}
HARDCopy?

**Related Commands** *WAI, *CLS

**Arguments**
START sends a block of data representing the current screen image to the requested port. The data sent is in the image format specified by the SAVe:IMAGe:FILEFormat command and the compression level is controlled by the selected format (BMP and TIFF are uncompressed while PNG is
Commands Listed in Alphabetical Order

...compressed). The SAVE:IMAGE:INKSaver determines whether the data sent is in InkSaver mode.

**NOTE.** The Hardcopy command with no arguments is equivalent to pressing the “HardCopy” button on the front panel, when the data is sent to the printer. As there is no front panel equivalent of the HARDCOPY START, you can choose to save the images to a different location using the “SAVE” button.

**Examples**

HARDCOPY initiates a screen copy to the active printer.

**HARDCopy:ACTIVeprinter**

Sets or returns the currently active printer. When a hard copy operation is performed, the output will be sent to this printer. One of two methods of specifying the printer can be used: specifying an index value obtained from looking at the list of attached printers or by specifying the printer name.

<table>
<thead>
<tr>
<th>Group</th>
<th>Hard Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>HARDCopy:ACTIVeprinter {&lt;NR1&gt;</td>
</tr>
<tr>
<td></td>
<td>HARDCopy:ACTIVeprinter?</td>
</tr>
</tbody>
</table>

| Arguments | <NR1> is the index of the desired printer as returned from HARDCopy:PRINTER:LIST? |
|           | <name> is the name of the printer as specified in the printer list. This name is case sensitive and must be entered exactly as shown in the list. |

**HARDCopy:INKSaver**

Changes hard copy output to print traces and graticule on a white background while retaining waveform color information (except for channel 1, which prints as dark blue because yellow does not show up well and is difficult to see on a white background). This option can significantly reduce print time and quantities of ink required compared with WYSIWYG dark background images.

<table>
<thead>
<tr>
<th>Group</th>
<th>Hard Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>HARDCopy:INKSaver?</td>
</tr>
</tbody>
</table>
Commands Listed in Alphabetical Order

Arguments

**ON** or **<NR1> ≠ 0** sets the ink saver mode on.
**OFF** or **<NR1> = 0** sets the ink saver mode off.

Examples

HARDCOPY:INKSAVER ON will cause subsequent hard copy output to display the screen on a white background.

HARDCopy:LAYout

Sets or returns the page orientation for hard copy. If you set the layout to LANDscape, the printer will print hard copies in landscape mode where the long edge of the screen will print to the long edge of the sheet of paper. If you set the layout to PORTRait, the printer will print hard copies in portrait mode.

This command is not applicable for PictBridge hardcopies.

Group

Hard Copy

Syntax

HARDCopy:LAYout {PORTRait|LANDscape}

Arguments

PORTRait orients the screen image vertically on the printed page.
LANDscape orients the screen image horizontally on the printed page.

Examples

HARDCOPY:LAYOUT LANDSCAPE sets the hard copy page orientation to Landscape.

HARDCOPY:LAYOUT? might return :HARDCOPY:LAYOUT PORTRAIT indicating that the hard copy page orientation is set to portrait.

HARDCopy:PREVIEW (No Query Form)

Displays a preview of the current screen contents with the InkSaver palette applied.

Group

Hard Copy

Syntax

HARDCopy:PREVIEW {ON|OFF|<NR1>}

Arguments

ON or <NR1> ≠ 0 turns preview mode on.
OFF or <NR1> = 0 turns preview mode off.
HARDCopy:PRINTer:ADD (No Query Form)

Adds a network printer to the list of available printers. All three arguments must be present, but only one of server name or server IP address may be specified. An empty string can be used for blank arguments.

**Group**  
Hard Copy

**Syntax**  
HARDCopy:PRINTer:ADD <name>,<server>,<address>

**Arguments**  
- `<name>` is the name of the network printer queue.
- `<server>` is the host name of the print (LPR) server.
- `<address>` is the IP address of the print server.

HARDCopy:PRINTer:DELETE (No Query Form)

Removes a network printer from the list of available printers. The printer name is case sensitive.

**Group**  
Hard Copy

**Syntax**  
HARDCopy:PRINTer:DELETE <name>

**Arguments**  
- `<name>` is the name of the printer to be deleted.

HARDCopy:PRINTer:LIST? (Query Only)

Returns a list of currently attached printers.

**Group**  
Hard Copy

**Syntax**  
HARDCopy:PRINTer:LIST?

HARDCopy:PRINTer:REName (No Query Form)

Renames a network printer on the list of available printers, replacing the currently stored settings with the settings specified in this command. Four arguments must
be present, but the arguments may be empty strings if the value for a field is to be deleted.

**Group** Hard Copy

**Syntax**

```
HARDCopy:PRINTER:REName
<name>,<new_name>,<new_server>,<new_address>
```

**Arguments**

- `<name>` is the name of the printer to be deleted.
- `<new_name>` is the new name for this printer.
- `<new_server>` is the new print server for this printer.
- `<new_address>` is the new IP address for the server.

**HEADer**

Sets or returns the Response Header Enable State that causes the oscilloscope 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. This command does affect the Response Header Enable State of both the USBTMC and VXI-11 interfaces. Refer to the Introduction for additional information.

**Group** Miscellaneous

**Syntax**

```
HEADer {OFF|ON|<NR1>}

HEADer?
```

**Related Commands** VERBose

**Arguments**

- `OFF` sets the Response Header Enable State to false. This causes the oscilloscope 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 oscilloscope to include headers on applicable query responses. You can then use the query response as a command.
- `<NR1>` = 0 sets the Response Header Enable State to false; any other value sets this state to true.
Examples

HEADER OFF specifies that the oscilloscope omits headers on query responses, so that only the argument is returned.

HEADER? might return :HEADER 1 indicating that the oscilloscope is including headers on applicable query responses.

HISTogram? (Query Only)

This query-only command returns all histogram parameters; it queries the state of all histogram parameters that the user can set. This command is equivalent to selecting Waveform Histograms from the Measure menu.

Group

Histogram

Syntax

HISTogram?

Related Commands

MEASUrement:METHod

Examples

HISTOGRAM? might return the histogram parameters as :HISTOGRAM:BOXPCNT 30.0000,25.1000,70.0000,75.2000;DISPLAY LINEAR;STATE 1;FUNCTION VERTICAL;SIZE 2.0000;SOURCE CH1.

HISTogram:BOX

This command sets or returns the histogram box coordinates in terms of source waveform vertical and horizontal units.

Group

Histogram

Syntax

HISTogram:BOX <NR3>, <NR3>, <NR3>, <NR3>

HISTogram:BOX?

Related Commands

HISTogram:BOXPcnt

Arguments

<NR3> specifies the left position of the histogram box in source waveform horizontal units.

<NR3> specifies the top position of the histogram box in source waveform vertical units.
<NR3> specifies the right position of the histogram box in source waveform horizontal units.

<NR3> specifies the bottom position of the histogram box in source waveform vertical units.

**Examples**

```
HISTOGRAM:BOX 1E-9, 0.250, 2E-9, 0.500 defines the coordinates of the histogram box in source waveform coordinates.

HISTOGRAM:BOX? might return :HISTOGRAM:BOX 1.0000E-09, 0.2500, 2.0000E-09, 0.5000, which identifies the coordinates of the histogram box in source waveform coordinates.
```

**HISTogram:BOXPcnt**

This command sets or returns the histogram box coordinates in terms of percentages of the full screen extents of the source waveform. The arguments refer to the left (percent of the horizontal screen extent), top (percent of the vertical screen extent), right (percent of the horizontal screen extent), bottom (percent of the vertical screen extent). The valid range for these values is 0.0 to 100.0.

**Group**

Hisstogram

**Syntax**

```
HISTogram:BOXPcnt <NR2>, <NR2>, <NR2>, <NR2>
HISTogram:BOXPcnt?
```

**Related Commands**

HISTogram:BOX

**Arguments**

<NR3> specifies the left position of the histogram box in percentage coordinates. The default value is 20%

<NR3> specifies the top position of the histogram box in percentage coordinates. The default value is 80%

<NR3> specifies the right position of the histogram box in percentage coordinates. The default value is 80%

<NR3> specifies the bottom position of the histogram box in percentage coordinates. The default value is 20%

**Examples**

```
HISTOGRAM:BOXPcnt 30.0, 25.0, 70.0, 75.0 sets the left boundary of the histogram box to 30% of the graticule (3 divisions from the left edge); the top boundary to 25% of the graticule (0.25 × 10 = 2.5 divisions from the top edge); the right boundary to 70% of the graticule (7 divisions from the left edge); and
```
the bottom boundary to 75% of the graticule (0.75 × 10 = 7.5 divisions from
the top edge).

HISTOGRAM: BOXPCNT? might return :HISTOGRAM: BOXPCNT
30.0000, 25.1000, 70.0000, 75.2000.

**HISTogram:COUNt (No Query Form)**

This command (no query form) clears the count and statistics for the histogram
and the histogram source data. If the histogram is on, then counting restarts.

**Group**
Histogram

**Syntax**
HISTogram:COUNt RESET

**Related Commands**
HISTogram:BOX
HISTogram:BOXPCnt

**Arguments**
RESET

**Examples**
HISTOGRAM:COUNT RESET clears the count and statistics for the histogram and
the histogram source data.

**HISTogram:DATa? (Query Only)**

This query returns the histogram data when the :HISTogram:MODE is
HORIZONTAL or VERTICAL. If the mode is OFF, then no data is returned and an
error event is set.

The data values returned for this query represent the number of times the histogram
source waveform samples were coincident with a particular histogram bin.

For vertical histograms, this query returns 256 values, representing the number
of times the histogram source waveform samples were coincident with each of
the 256 digitizing levels. Of these 256 values, the first 2 and last 3 are always
0, as they represent digitizing levels that fall above and beneath the waveform
graticule, respectively.

For horizontal histograms, this query returns 1000 values, representing the
number of times the histogram source waveform samples were coincident with
each horizontal pixel column. The time of occurrence for each of the horizontal
bins can be derived using the HISTogram:STARt? and HISTogram:END? queries.
**Group**  
Histogram

**Syntax**  
HIStogram:DATa?

**Related Commands**  
HIStogram:BOX  
HIStogram:BOXPcnt  
HIStogram:COUNt  
HIStogram:DISplay  
HIStogram:MODe  
HIStogram:SOUrce

**Examples**  
HIStogram:DATa? might return :HIStogram:DATa  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,0,0,0,1245,933,903,1003,838,782,764,697,  
687,620,582,566,530,595,557,552,518,524,606,  
503,526,470,498,554,534,487,546,548,504,554,  
513,566,639,508,643,704,662,769,785,806,922,  
960,1153,1336,1478,1960,4515,17598,52184,  
67040,50417,12885,4104,2382,1866,1699,1428,  
1288,1271,1140,975,1024,894,841,841,678,841,  
804,685,713,746,729,644,640,787,768,668,671,  
661,745,755,692,882,772,802,862,770,854,799,  
969,1102,1080,1040,1272,1329,1394,1760,1428,  
2418,3313,4475,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.

**HIStogram:DISplay**  
This command sets the scaling of the histogram data display to be the count of each histogram bin or the log of that count. The default scaling is linear.

**Group**  
Histogram

**Syntax**  
HIStogram:DISplay {OFF|LOG|LINEAr}  
HIStogram:DISplay?

**Related Commands**  
HIStogram:MODe
Arguments  

- `<LOG>` sets the histogram display to logarithmic scaling
- `<LINEAR>` sets the histogram display to linear scaling. This is the default setting.

Examples  

- `HISTOGRAM:DISPLAY LOG` sets the histogram scaling to be the log count of each histogram bin.
- `HISTOGRAM:DISPLAY?` might return `:HISTOGRAM:DISPLAY LOG` indicating that the current histogram display scaling is the log count of each histogram bin.

**HISTogram:END? (Query Only)**

Returns the time (horizontal) or vertical units value (vertical) of the last bin where the histogram data ends. The `HISTogram:MODE` must be either `HORizontal` or `VERTical` for a value to be returned. If the mode is `OFF`, an error event is set and nothing is returned.

If the `HISTogram:MODE` is `HORizontal`, the value returned is the time of the right bin. If the `HISTogram:MODE` is `VERTical` the value returned is the vertical units value of the bottom bin. The returned value is an `<NR3>`.

Group  

Histogram

Syntax  

`HISTogram:END?`

Related Commands  

- `HISTogram:MODE`
- `HISTogram:STARt?`

Returns  

- `<NR3>`

Examples  

- `HISTogram:END?` might return `:HISTOGRAM:END 1.6000E-05` indicating that the last bin is at 16 μs.

**HISTogram:MODE**

This command selects the type of histogram to create or disables the histogram display. The query form either returns the current histogram type or that the histogram display is disabled.

Group  

Histogram
Syntax  

HIS trogram:MODE {HORIZONT al|VERTical|OFF}  

HIS trogram:MODE?

Related Commands  

HIS trogram:DISplay

Arguments  

HORIZONT al enables a horizontally positioned histogram that shows time distribution.  

VERTical enables a vertically positioned histogram that shows a voltage distribution, or another distribution such as amperes.  

OFF disables the collection of the histogram measurement.

Examples  

HISTOGRAM:MODE HORIZONTAL sets the type of histogram created to horizontal, which displays a horizontally positioned histogram that shows time distribution.  

HISTOGRAM:MODE? might return :HISTOGRAM:MODE OFF, indicating that the histogram display is disabled.

HIS trogram:SOURce

This command sets or queries which source will be compared against the histogram box when the histogram testing is enabled.

Group  

Histogram

Syntax  

HIS trogram:SOURce {CH<x>|MATH|REF<x>}

HIS trogram:SOURce?

Related Commands  

HIS trogram:DISplay

Arguments  

CH<x> selects a channel waveform as the source for the histogram. The x variable can be expressed as an integer ranging from 1 through 4.  

MATH selects the math waveform as the source for the histogram.  

REF<x> selects a reference waveform as the source for the histogram. The x variable can be expressed as an integer ranging from 1 through 4.

Examples  

HISTOGRAM:SOUR ce CH1 enables the Channel 1 waveform to be compared against the histogram box.
HISTOGRAM:SOURCE? might return :HISTOGRAM:SOURCE CH1, indicating that
the waveform for Channel 1 is the source for the histogram.

**HISTogram:STARt? (Query Only)**

Returns the time (horizontal) or vertical units value (vertical) of the first bin where
the histogram data starts. The HISTogram:MODE must be either HOrizontal or VERTical for a value to be returned. If the mode is OFF, an error event is set
and nothing is returned.

If the HISTogram:MODE is HOrizontal, the value returned is the time of the left
bin. If the HISTogram:MODE is VERTical the value returned is the vertical units
value of the top bin. The returned value is an <NR3>.

**Group**
Histogram

**Syntax**
HISTogram:STARt?

**Related Commands**
HISTogram:END?
HISTogram:MODE

**Returns**
<NR3> showing the returned value.

**Examples**
HISTOGRAM:STARt? might return :HISTOGRAM:START 2.0000E-06
indicating that the first bin is at 2 μs.

**HORizontal? (Query Only)**

Returns all settings for the horizontal commands.

**Group**
Horizontal

**Syntax**
HORizontal?

**Examples**
HORIZONTAL? might return the following horizontal settings
:HORIZONTAL:DELAY:MODE 1;TIME 0.0000;:HORIZONTAL:MAIN:SCALE 20.0000E-9;SAMPLERATE 5.0000E+9;UNITS "s";UNITS:STRING "s";:HORIZONTAL:SAMPLERATE 5.0000E+9;RECORDLENGTH 1000000
**Horiz ontal:Del ay:Mode**

Sets or returns the horizontal delay mode.

**Group**
Horizontal

**Syntax**

```
HORizontal:DELay:MODe {OFF|ON|<NR1>}
HORizontal:DELay:MODe?
```

**Related Commands**
HORizontal:PO Sition

**Arguments**

- **OFF** sets the Horizontal Delay Mode to off. This causes the HORizontal:POSE tion 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.
- **<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:DELA Y: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.

---

**Horiz ontal:Delay:Time**

Sets or returns the horizontal delay time (position) that is used when delay is on (the default mode).

**Group**
Horizontal

**Syntax**

```
HORizontal:DELay:TIme <NR3>
HORizontal:DELay:TIme?
```

**Related Commands**
HORizontal:PO Sition
Arguments

NR3 is the delay in seconds.

Examples

HORizontal:DELa:y:TIME 0.3 sets the delay of acquisition data so that the resulting waveform is centered 300 ms after the trigger occurs.

HORizontal:DIGital:RECORDlength:MAGnivu? (Query Only)

Returns the record length of the MagniVu digital acquisition.

Group

Horizontal

Syntax

HORizontal:DIGital:RECORDlength:MAGnivu?

HORizontal:DIGital:RECORDlength:MAIN? (Query Only)

Returns the record length of the main digital acquisition.

Group

Horizontal

Syntax

HORizontal:DIGital:RECORDlength:MAIN?

HORizontal:DIGital:SAMPLERate:MAGnivu? (Query Only)

Returns the sample rate of the MagniVu digital acquisition.

Group

Horizontal

Syntax

HORizontal:DIGital:SAMPLERate:MAGnivu?

HORizontal:DIGital:SAMPLERate:MAIN? (Query Only)

Returns the sample rate of the main digital acquisition.

Group

Horizontal

Syntax

HORizontal:DIGital:SAMPLERate:MAIN?
**HORizontal:POSition**

Sets or returns the horizontal position, in percent, that is used when delay is off. If 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 on, this command stores a new horizontal position that is used when Horizontal Delay Mode is turned off.

**Group**  
Horizontal

**Syntax**  

```
HORizontal:POSition <NR3>
HORizontal:POSition?
```

**Related Commands**  
HORizontal:DELay:TIMe

**Arguments**  

<NR3> is the horizontal position expressed as the percentage of the waveform acquired before the trigger.

**Examples**  

```
HORIZONTAL:POSITION 50 sets the horizontal position to 50%.
HORIZONTAL:POSITION? might return :HORIZONTAL:POSITION 100 indicating that the horizontal position is set to 100%.
```

**HORizontal:PREViewstate? (Query Only)**

Returns a boolean value to indicate whether the display system is in the preview state.

**Group**  
Horizontal

**Syntax**  

```
HORizontal:PREViewstate?
```

**Returns**  

<NR1> = 1 if the display system is in the preview state.
<NR1> = 0 if the display system is not in the preview state.

**HORizontal:RECOrdlength**

Sets the horizontal record length. The query form of this command returns the current horizontal record length.
**Group**: Horizontal

**Syntax**
- `HORizontal:RECORDlength <NR1>`
- `HORizontal:RECORDlength?`

**Arguments**
- `<NR1>` represents the supported values for horizontal record lengths, which are: 1000, 10000, 100000, 1000000, or 10000000.

**Examples**
- `HORIZONTAL:RECORDLENGTH 10000` specifies that 10000 data points will be acquired for each record.
- `HORIZONTAL:RECORDLENGTH?` might return `:HORIZONTAL:RECORDlength 1000` indicating that the horizontal record length is equal to 1000 data points.

**HORizontal:SAMPLERate**

Sets or returns the current horizontal sample rate.

**Group**: Horizontal

**Syntax**
- `HORizontal:SAMPLERate <NR3>`
- `HORizontal:SAMPLERate?`

**Arguments**
- `<NR3>` is the rate in samples per second.

**HORizontal:SCAle**

Sets or returns the time base horizontal scale.

**Group**: Horizontal

**Syntax**
- `HORizontal:SCAle <NR3>`
- `HORizontal:SCAle?`

**Arguments**
- `<NR3>` specifies the range from 400 ps to 1000 s, depending on the record length.
Examples

HORIZONTAL:SCALE 2E-6 sets the main scale to 2µs per division.

HORIZONTAL:SCALE? might return :HORIZONTAL:MAIN:SCALE 2.0000E-06 indicating that the main scale is currently set to 2 µs per division.

ID? (Query Only)

Returns identifying information about the oscilloscope and related firmware.

Group

Miscellaneous

Syntax

ID?

Related Commands

*IDN?

Examples

ID? might return TEK/DPO4034,CF:91.1CT,FV:v1.0000. This indicates the oscilloscope model number, configured format, and firmware version number.

*IDN? (Query Only)

Returns the oscilloscope identification code.

Group

Miscellaneous

Syntax

*IDN?

Related Commands

ID?

Examples

*IDN? might return :TEKTRONIX,DPO4034,SN123456789,CF:91.1CT FV:v1.00000 indicating the oscilloscope model number, serial number, configured number, and firmware version number.

LANGuage

Sets or returns the user interface display language. This command only affects the oscilloscope displayed language. Remote commands and their responses are always in English.
Group    Miscellaneous

Syntax    LANGUAGE
{ENGLISH|FRENCH|GERMAN|ITALIAN|SPANISH|PORTUGUESE|JAPANESE|
KOREAN|RUSSIAN|SIMPLIFIEDCHINESE|TRADITIONALCHINESE}

Examples    LANGUAGE? might return :LANGUAGE ENGLISH.

LOCk

Enables or disables all front-panel buttons and knobs. There is no front panel equivalent.

When the front panel is locked, neither the FPanel:PRESS nor the FPanel:TURN commands will work. They will not generate an error event either. You can work around this 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

Arguments    ALL disables all front-panel controls.

            NONE enables all front-panel controls. This is equivalent to the UNLock ALL command.

Examples    LOCk ALL locks the front-panel controls.

            LOCk? might return :LOCk NONE indicating that the front-panel controls are enabled by this command.

*LRN? (Query Only)

Returns the commands that list the oscilloscope settings except for configuration information for the calibration values, the WFMinpre? query, and the WFMOutpre? query. This query allows you to record or "learn" the current
oscilloscope settings. You can use these commands to return the oscilloscope 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 a long response, part of which could be as follows:

```
:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV
INFINITE;NUMAVG 16;SAMPLINGMODE RT.;:HEADER 1;:LOCK
NONE;:LANGUAGE ENGLISH;:VERBOSE 1;:ALIAS:STATE
0;:DISPLAY:COLOR:PALETTE NORMAL;:DISPLAY:STYLE:DOTONLY
0;:DISPLAY:PERSISTENCE 0.0000;CLOCK 1;GRATICULE
FULL;INTENSITY:WAVEFORM 30;GRATICULE 75;BACKLIGHT
HIGH;:HARDCOPY :INKSAVER OFF;LAYOUT LANDSCAPE;PREVIEW
0;:SAVE:IMAGE:FILEFORMAT BMP;:SAVE:WAVEFORM:FILEFORMAT
INTERNAL;:SAVE:ASSIGN:TYPE SETUP;:TRIGGER:A:MODE
AUTO;TYPE EDGE;LE VEL 20.0000E-3;LEVEL:CH1
20.0000E-3;CH2 0.0000;CH3 0.0000;CH4 0.0000;:TRIGGER:A:
UPPERTHRESHOLD:CH1 1.4000;CH2 800.0000E-3;CH3
800.0000E-3;CH4 800.0000E-3;:TRIGGER:A:LOWERTHRESHOLD:
CH1 20.0000E-3;CH2 0.0000;CH3 0.0000;CH4 0.0000;:TRIGGER:A:
OLDOFF:TIME 20.0000E-9;:TRIGGER:A:EDGE:SOURCE
CH1;COUPLING DC;SLOPE RISE;:TRIGGER:A:LOGIC:CLASS
SETHOLD;FUNCTION AND;THRESHOLD:CH1 20.0000E-3;CH2
0.0000;CH3 0.0 00;CH4 0.0000;:TRIGGER:A:LOGIC:INPUT:CH1
X;CH2 X;CH3 X;CH4 X;CLOCK:SOURCE NONE; EDGE
RISE;:TRIGGER:A:LOGIC:PATTERN:INPUT:CH1 X;CH2
X;CH3 X;CH4 X;:TRIGGER:A:LOGIC:PATTERN:WHEN
TRUE;WHEN:LESSLIMIT 4.0000E-9;LIMIT
4.0000E-9;:TRIGGER:A:SETHOLD:CLOCK:SOURCE CH1;EDGE
RISE;THRESHOLD 20.0000E-3;:TRIGGER:A:SETHOLD:DATA:SOURCE
CH2;THRESHOLD 0.0000;:TRIGGER:A:SETHOLD:HOLDTIME
4.0000E-9;SETTIME 4.0000E-9; :TRIGGER:A:PULSE:CLASS
TRANSITION;:TRIGGER:A:PULSEWIDTH:SOURCE
CH1;POLARITY POSITIVE;WHEN LESSTHAN:WIDTH
4.0000E-9;:TRIGGER:A:RUNT:SOURCE CH1;POLARITY POSITIVE; WHEN
OCCURS:WIDTH 4.0000E-9;:TRIGGER:A:TRANSITION:SOURCE CH1
...```
MARK

Moves to the next or previous reference mark on the waveform. Returns the current mark position.

Group  Mark

Syntax  MARK {NEXT|PREVIOUS}  
        MARK?

Arguments  NEXT moves to the next reference mark on the right.  
            PREVIOUS moves to the next reference mark on the left.

MARK:CREATE (No Query Form)

Creates a mark on a specified waveform or all waveforms in a column.

Group  Mark

Syntax  MARK:CREATE {CH<x>|MATH|B<x>|REF<x>|DIGital|COLUMN}

Arguments  CH<x> creates the mark on a channel waveform, where <x> is the channel number.  
            MATH creates the mark on the math waveform.  
            B<x> creates the mark on a bus waveform, where <x> is the bus number.  
            REF<x> creates the mark on a reference waveform, where <x> is the reference waveform number.  
            DIGital creates the mark on a digital waveform. (An error will result if no digital channel is turned on.)  
            COLUMN creates marks on all waveforms in the current zoom pixel column.

MARK:DELETE (No Query Form)

Deletes a mark on a particular waveform, all waveforms in a column, the selected mark, or all marks.

Group  Mark
Syntax  
MARK:DELETE \{CH<x>|MATH|B<x>|REF<x>|DIGital|COLUMN\}

Arguments  
CH<x> deletes the mark on a channel waveform, where <x> is the channel number.
MATH deletes the mark on the math waveform.
B<x> deletes the mark on a bus waveform, where <x> is the bus number.
REF<x> deletes the mark on a reference waveform, where <x> is the reference waveform number.
DIGital deletes all marks on all digital channels.
COLUMN deletes marks on all waveforms in the current zoom pixel column.

MARK:FREE? (Query Only)

Returns how many marks are available for use.

Group  Mark

Syntax  MARK:FREE?

MARK:SELECTed:END? (Query Only)

Returns the end of the selected mark, 0 to 100% of the waveform.

Group  Mark

Syntax  MARK:SELECTed:END?

MARK:SELECTed:FOCUS? (Query Only)

Returns the focus of the selected mark, 0 to 100% of the waveform.

Group  Mark

Syntax  MARK:SELECTed:FOCUS?
MARK:SELECTed:MARKSINCOUm? (Query Only)

Returns the number of marks in the current zoom pixel column.

**Group**  
Mark

**Syntax**  
MARK:SELECTed:MARKSINCOUm?

MARK:SELECTed:OWNer? (Query Only)

Returns the owner of the selected mark.

**Group**  
Mark

**Syntax**  
MARK:SELECTed:OWNer?

**Returns**  
<QString> is the owner of the mark.

**Examples**  
MARK:SELECTed:OWNer? might return: USER, SEARCH1

MARK:SELECTed:SOURCE? (Query Only)

Returns the source waveform for the selected mark.

**Group**  
Mark

**Syntax**  
MARK:SELECTed:SOURCE?

MARK:SELECTed:START? (Query Only)

Returns the starting point of the selected mark, 0 to 100% of the waveform.

**Group**  
Mark

**Syntax**  
MARK:SELECTed:START?
**MARK:SELECTed:STATE? (Query Only)**

Returns the on or off state of the selected mark. The selected mark is at or near the center of the screen. If you press the front-panel Set/Clear button, this mark will disappear.

**Group**  
Mark

**Syntax**  
MARK:SELECTed:STATE?

**MARK:SELECTed:ZOOM:POSITION? (Query Only)**

Returns the position of the selected mark, 0 to 100% of the zoom overview window.

**Group**  
Mark

**Syntax**  
MARK:SELECTed:ZOOM:POSITION?

**MARK:TOTa1? (Query Only)**

Returns how many marks are currently in use.

**Group**  
Mark

**Syntax**  
MARK:TOTa1?

**{MATH|MATH1}:LABel**

Sets or queries the waveform label for the math waveform.

**Group**  
Math

**Syntax**  
{MATH|MATH1}:LABel <QString>  
{MATH|MATH1}:LABel?

**Arguments**  
<QString> is the quoted string used as the label for the math waveform.
Examples

MATH:LABEL "Output" sets the label for the math waveform to Output.

MATH:LABEL? might return MATH:LABEL "Sum of channel 1 and channel 2" indicating the current label for the math waveform.

MATH[1]? (Query Only)

Returns the definition of the math waveform. The returned data depends on the setting of the MATH[1]:TYPE command.

Group
Math

Syntax
MATH[1]?

Related Commands
MATH[1]:TYPE

Examples

MATH? or MATH1? might return :MATH:TYPE DUAL;DEFINE
"CH1+CH2";VERTICAL:SCALE 100.0000E-3;POSITION 0.0000;UNITS
"V";:MATH:HORIZONTAL:SCALE 4.0000E-6;POSITION 50.0000;UNITS
"s";:MATH:SPECTRAL:MAG DB;WINDOW HANNING

MATH[1]:DEFine

Sets or returns the current math function as a text string.

Dual math is defined if the string is of the form <wfm> <operation> <wfm>, where the <wfm>s are any combination of live channels or reference waveforms, <operation> is any of +, -, *, or /, and the MATH[1]:TYPE is DUAL.

FFT math is defined if the string is in the form FFT(<wfm>), where <wfm> is any live channel or reference waveform, and the MATH[1]:TYPE is FFT.

Advanced math is defined if the contents of the string can be parsed by the advanced math parser without errors and the MATH[1]:TYPE is ADVanced.

On the front panel, the Dual Wfm Math, FFT and Advanced Math menus contain controls that allow building equivalent math expressions to those described above.

Group
Math

Syntax
MATH[1]:DEFine <QString>
MATH[1]:DEFine?
Related Commands

**MATHVAR:VAR<x>, MATH[1]:TYPE**

**Arguments**

**<QString>** quoted string argument is the mathematical expression that defines the waveform.

### Table 2-42: Advanced Math expression elements

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH1-CH4, REF1-REF4</td>
<td>Specifies a waveform data source.</td>
</tr>
<tr>
<td>FFT(, INTG(, DIFF(</td>
<td>Executes a Fast Fourier Transform, integration, or differentiation operation on the expression that follows. The FFT operator must be the first (left-most) operator in an expression. All these operations must end with a right parenthesis.</td>
</tr>
<tr>
<td>AMPlitude(, AREa(, BURst(, CAREa(, CMEan(, CRMs(, DELay(, FALL(, FREQuency(, HIGH(, LOW(, MAXimum(, MEAN(, MINImum(, NDUty(, NOVershoot(, NWIdth(, PDUTy(, PERIod(, PHAse(, PK2pk(, POVershoot(, PWidth(, RISe(, RMS(, !)</td>
<td>Executes the selected measurement operation on the waveform (active or reference) that follows. All these operations must end with a right parenthesis.</td>
</tr>
<tr>
<td>LOG(, EXP(, SQRT(, SINE(, COSINE(, TANGENT(</td>
<td>Executes trigonometric and other functions. All these operations must end with a right parenthesis.</td>
</tr>
<tr>
<td>VAR1, VAR2</td>
<td>Adds the user-defined variable to the expression. Refer to the MATHVAR&lt;x&gt; command.</td>
</tr>
<tr>
<td>+,-,*,/</td>
<td>Executes an addition, subtraction, multiplication, or division operation on the following expression. + and - are also unary; use - to negate the expression that follows.</td>
</tr>
<tr>
<td>&lt;, &gt;, &lt;=, &gt;=, ==, ≠,</td>
<td></td>
</tr>
<tr>
<td>()</td>
<td>Parentheses provide a way to control evaluation order in an expression. The comma is used to separate the &quot;from&quot; and &quot;to&quot; waveforms in Delay and Phase measurement operations.</td>
</tr>
<tr>
<td>1-0, ., E</td>
<td>Specifies a numeric value in (optional) scientific notation.</td>
</tr>
</tbody>
</table>

**Examples**

MATH1:DEFINE " CH1+CH2" adds the Ch 1 waveform and Ch 2 waveform, storing the results in Math 1.

MATH:DEFINE? might return :MATH1:DEFINE "CH2*REF2" as the expression that defines Math 1.
**MATH[1]:HORizontal:POSition**

Sets or returns the math horizontal display position for FFT or (non-live) math reference waveforms.

**Group**  
Math

**Syntax**  
MATH[1]:HORizontal:POSition <NR3>  
MATH[1]:HORizontal:POSITION?

**Arguments**  
<NR3> is the % of the math waveform that precedes center screen. It can vary from 0.0 to 100.0.

**Examples**  
MATH:HORIZONTAL:POSITION 10 sets the horizontal position to 10% pretrigger

**MATH[1]:HORizontal:SCAle**

Sets or returns the math horizontal display scale for FFT or for dual math waveforms that have source waveforms that are reference waveforms. The horizontal scale of a dual math waveform with a channel source waveform is set through the HORizontal:SCAle command.

**Group**  
Math

**Syntax**  
MATH[1]:HORizontal:SCAle <NR3>  
MATH[1]:HORizontal:SCAle?

**Arguments**  
<NR3> is the math horizontal scale in seconds.

**Examples**  
MATH:HORIZONTAL:SCALE? might return MATH:HORIZONTAL:SCALE 2.0E-4 indicating that the math horizontal scale is 200 μ

**MATH[1]:HORizontal:UNIts**

Returns the math waveform horizontal measurement unit value.

**Group**  
Math
Commands Listed in Alphabetical Order

**Syntax**

MATH[1]:HORIZONTAL:UNITS?

**Examples**

MATH:HORIZONTAL:UNITS? might return MATH:HORIZONTAL:UNITS "?"
indicating that the math horizontal unit label for unknown values is the default question mark unit.

---

**MATH[1]:SPECTral:MAG**

Sets or returns the units of the Spectral Magnification function in the math string.

**Group**

Math

**Syntax**

MATH[1]:SPECTral:MAG {LINEAR|DB}
MATH[1]:SPECTral:MAG?

**Arguments**

LINEAR sets the SpectralMag units to linear.
DB sets the SpectralMag units to decibels.

**Examples**

MATH1:SPECTRAL:MAG DB sets the SpectralMag units for Math1 to decibels.
MATH1:SPECTRAL:MAG? might return :MATH1:SPECTRAL:MAG DB indicating that the SpectralMag units for Math1 are set to decibels.

---

**MATH[1]:SPECTral:WINdow**

Sets or returns the window function for the spectral analyzer input data for the specified math waveform. 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.

**Group**

Math

**Syntax**

MATH[1]:SPECTral:WINdow
{RECTangular|HAMming|HANning|BLAckmanharris}
MATH[1]:SPECTral:WINdow?
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.

Examples

MATH1:SPECTRAL:WINDOW HANNING applies a Hanning window to the spectral analyzer input data.

MATH1:SPECTRAL:WINDOW? might return :MATH1:SPECTRAL:WINDOW HAMMING indicating that the window function used to multiply the spectral analyzer input data is the Hamming window.

**MATH[1]:TYPe**

Sets or returns the math waveform mode type.

**Group**

Math

**Syntax**

MATH[1]:TYPe {ADVanced|DUAL|FFT}

MATH[1]:TYPe?

**Arguments**

ADVanced sets the math waveform mode to advanced math.

DUAL sets the math waveform mode to dual waveform math.

FFT sets the math waveform mode to FFT math.

**Examples**

MATH:TYPE FFT sets the math waveform mode to FFT.

MATH:TYPE FFT; :MATH:DEFINE "FFT( CH1 )" sets the math type to FFT and displays an FFT waveform of the channel 1 waveform, using the current FFT scale and window settings.

MATH:TYPE ADVANCED; :MATH:DEFINE "INTG(REF1*CH3)+DELAY(CH1,CH2)" sets the math type to FFT and displays an advanced math waveform that is the integration of the product of REF1 and CH3 plus the result of the delay measurement between channel 1 and 2.

**MATH[1]:VERTical:POSition**

Sets or returns the vertical position of the currently selected math type.
**Group** Math

**Syntax**

MATH[1]:VERTical:POSition <NR3>
MATH[1]:VERTical:POSition?

**Related Commands**

CH<x>:POSition, REF<x>:VERTical:POSition

**Arguments**

<NR3> is the desired position in divisions from the center graticule.

**Examples**

MATH1:VERTICAL:POSITION 1.3E+00 positions the Math 1 input signal 1.3 divisions higher than a position of 0.

MATH1:VERTICAL:POSITION? might return :MATH1:VERTICAL:POSITION -1.3000E+00 indicating that the current position of Math 1 is 1.3 divisions below the center graticule.

---

**MATH[1]:VERTical:SCAle**

Sets or returns the vertical scale of the currently selected math type.

**Group** Math

**Syntax**

MATH[1]:VERTical:SCAle <NR3>
MATH[1]:VERTical:SCAle?

**Related Commands**

CH<x>:SCAle, REF<x>:VERTical:SCAle

**Arguments**

<NR3> is the scale-per-division in the current math vertical units. The range is from 1.0E-12 through 500.0E+12.

**Examples**

MATH1:VERTICAL:SCALE 100E-03 sets the Math scale to 100 mV per division.

MATH:VERTICAL:SCALE? might return :MATH:VERTICAL:SCALE 1.0000E+00 indicating that the current scale setting of Math is 1 V per division.

---

**MATH[1]:VERTical:UNIts**

Returns the math waveform vertical measurement unit value.
**Group** Math

**Syntax** MATH[1]:VERTical:UNIts?

**Examples** MATH:VERTICAL:UNITS? might return MATH:VERTICAL:UNITS "joules" indicating that the math vertical unit label for unknown values is joules.

**MATHVAR? (Query Only)**

Queries both numerical values you can use within math expressions.

**Group** Math

**Syntax** MATHVAR?

**Related Commands** MATHVAR:VAR<x>, MATH[1]:DEFine

**Returns** <NR3> are the stored numerical values.

**Examples** MATHVAR? returns the values of all variables stored in locations 1 through 2.

**MATHVAR:VAR<x>**

Sets or returns one of two different numerical values you can use within math expressions. These values can range from -10.0e-18 to 1.0e+15; the default values are 0.0. <x> specifies the location, 1 or 2, in which you can store values. Stored math variables can be referenced within math expressions as VAR1 and VAR2.

For example, the following command defines MATH1 as the product of Channel 1 and math variable 1: MATH1:DEFINE "CH1 * VAR1".

**Group** Math

**Syntax** MATHVAR:VAR<x> <NR3>

**Related Commands** MATHVAR:VAR<x>, MATH[1]:DEFine
Arguments

<NR3> specifies the numerical value to be stored in location x <1 through 2>.

Examples

MATHVAR:VAR2 -2.43E-5 stores the value -2.43e-5 in the second math variable location.


MEASUrement? (Query Only)

Returns all measurement parameters.

Group
Measurement

Syntax
MEASUrement?

Examples

MEASUREMENT? might return :MEASUREMENT:IMMED:DELAY:DIRECTION FORWARDS;EDGE1 RISE;EDGE2 RISE;:MEASUREMENT:IMMED:TYPE PERIOD;UNITS "s";SOURCE1 CH1;SOURCE2 CH2;:MEASUREMENT:MEAS1:DELAY:DIRECTION FORWARDS;EDGE1 RISE;EDGE2 RISE;:MEASUREMENT:MEAS1:STATE 1;TYPE FREQUENCY;UNITS "Hz";SOURCE1 CH1;SOURCE2 CH2;COUNT 0;MAXIMUM 0.0000;MEAN 0.0000;MINIMUM 0.0000;STDDEV 0.0000;:MEASUREMENT:MEAS2:DELAY:DIRECTION FORWARDS;EDGE1 RISE;EDGE2 RISE;:MEASUREMENT:MEAS2:STATE 1;TYPE PERIOD;UNITS "s";SOURCE1 CH1;SOURCE2 CH2;COUNT 0;MAXIMUM 0.0000;MEAN 0.0000;MINIMUM 0.0000;STDDEV 0.0000;:MEASUREMENT:MEAS3:DELAY:DIRECTION FORWARDS;EDGE1 RISE;EDGE2 RISE;:MEASUREMENT:MEAS3:STATE 1;TYPE PK2PK;UNITS "V";SOURCE1 CH1;SOURCE2 CH2;COUNT 0;MAXIMUM 0.0000;MEAN 0.0000;MINIMUM 0.0000;STDDEV 0.0000;:MEASUREMENT:MEAS4:DELAY:DIRECTION FORWARDS;EDGE1 RISE;EDGE2 RISE;:MEASUREMENT:MEAS4:STATE 0;TYPE PERIOD;UNITS "s";SOURCE1 CH1;SOURCE2 CH2;COUNT 0;MAXIMUM 0.0000;MEAN 0.0000;MINIMUM 0.0000;STDDEV 0.0000;:MEASUREMENT:METHOD AUTO;REFLEVEL:METHOD PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2 0.0000;:MEASUREMENT:INDICATORS:STATE OFF;NUMHORZ 0;NUMVERT 0;HORZ1 99.0000E+36;HORZ2 99.0000E+36;HORZ3 99.0000E+36;HORZ4 99.0000E+36;VERT1 99.0000E+36;VERT2 99.0000E+36;VERT3 99.0000E+36;VERT4 99.0000E+36;:MEASUREMENT:STATISTICS:MODE OFF;WEIGHTING 32;:MEASUREMENT:GATING SCREEN.
MEASUrement:CLEARSNapshot (No Query Form)

Removes the measurement snapshot display.

**Group** Measurement

**Syntax** MEASUrement:CLEARSNapshot

**Related Commands** CLEARMenu

MEASUrement:GATing

Specifies or returns the measurement gating setting.

**Group** Measurement

**Syntax** MEASUrement:GATing {OFF|SCREEN|CURSOR}

**MEASUrement:GATing?**

**Arguments**

- **OFF** turns off measurement gating (full 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.

**Examples**

MEASUREMENT:GATING CURSOR turns on measurement gating using the cursors as limits.

MEASUREMENT:GATING? might return :MEASUREMENT:GATING CURSOR indicating that measurements are limited to the portion of the waveform between the vertical bar cursors.

MEASUrement:IMMed? (Query Only)

Returns all immediate measurement setup parameters.

**Group** Measurement

**Syntax** MEASUrement:IMMed?
**MEASUrement:IMMed:DELay? (Query Only)**

Returns information about the immediate delay measurement. This command is equivalent to viewing the delay measurement settings on the measurement readout.

**Group**  
Measurement

**Syntax**  
MEASUrement:IMMed:DELay?

**Examples**  
MEASUREMENT:IMMED:DELAY? might return  
:MEASUREMENT:IMMED:DELAY:DIRECTION FORWARDS; EDGE1 RISE; EDGE2 RISE; :MEASUREMENT:IMMED:TYPE PERIOD; UNITS "s"; SOURCE1 CH1; SOURCE2 CH2

**MEASUrement:IMMed:DELay:DIRection**

Sets or returns the starting point and direction that determines the delay "to" edge when taking an immediate delay measurement.

**NOTE.** Use the MEASUrement:IMMed:SOUrce<x> command to specify the delay "to" waveform.

**Group**  
Measurement

**Syntax**  
MEASUrement:IMMed:DELay:DIRection {BACKwards|FORwards}  
MEASUrement:IMMed:DELay:DIRection?

**Related Commands**  
MEASUrement:IMMed:SOUrce<x>

**Arguments**  
BACKwards starts the search at the end of the waveform and looks for the last rising or falling edge in the waveform.
FORwards starts the search at the beginning of the waveform and looks for the first rising or falling edge in the waveform.
Examples

MEASUREMENT:IMMED:DELAY:DIRECTION FORWARDS starts searching from the beginning of the waveform record and looks for the first rising or falling edge.

MEASUREMENT:IMMED:DELAY:DIRECTION? might return :MEASUREMENT:IMMED:DELAY:DIRECTION BACKWARDS indicating that searching begins at the end of the waveform record and looks for the last rising or falling edge.

MEASUREMENT:IMMed:DELay:EDGE<x>

Sets or returns the slope of the edge the oscilloscope uses for the delay "from" or "to" waveform when taking an immediate delay measurement.

Group
Measurement

Syntax
MEASUREMENT:IMMed:DELAY:EDGE<x> {FALL|RISe}
MEASUREMENT:IMMed:DELAY:EDGE<x>?

Related Commands
MEASUREMENT:IMMed:SOUrce<x>

Arguments
<x> specifies which waveform to use, where <x> = 1 is the "from" waveform, and <x> = 2 is the "to" waveform.

FALL specifies the falling edge.

RISe specifies the rising edge.

Examples
MEASUREMENT:IMMED:DELAY:EDGE1 RISE specifies that the "from" waveform rising edge be used for the immediate delay measurement.

MEASUREMENT:IMMED:DELAY:EDGE1? returns either RISE or FALL.

MEASUREMENT:IMMed:SOUrce<x>

For SOURce1: Sets or returns the source for all single channel measurements. For delay or phase measurements, sets or returns the waveform to measure "from".

For SOUrc2: Sets or returns the waveform to measure "to" when taking a delay measurement or phase measurement.

NOTE. If you do not specify a numerical suffix, the source is assumed to be SOURce1.
Group Measurement

Syntax MEASUrement:IMMed:SOURce<x> {CH1|CH2|CH3|CH4|MATH|D<x>|HIStogram}
MEASUrement:IMMed:SOURce<x>?

Arguments CH1–CH4 or MATH is the source waveform.

MATH is the math waveform.

D<X> is a digital waveform. The x variable can be expressed as an integer, where x is the digital channel number. Digital channels are available only on MSO models.

HIStogram indicates the histogram as the object to be measured. HIStogram only applies to SOUrce1; it is not available for SOUrce2.

MEASUrement:IMMed:TYPe

Sets or returns the immediate measurement type.

Digital channel measurements do not have a user-settable midRef threshold. If you specify a digital channel measurement that is not available on MSO models, measurement error 2200: measurement system error occurs and 9.9e37 is returned.

Group Measurement

Syntax MEASUrement:IMMed:TYPe {AMPlitude|AREa|BURst|CArea|CMEan|CRMs|DElay|FALL|FREQuency |HIGH|HITS|LOW|MAXimum|MEAN|MEDian|MINImum|NDUty|NEDGECount|NOVershoot|NPULSECount|NWIdth|PEAKHits|PEDGECount|PDuty |PERIod|PHAse|PK2Pk|POVershoot|PPULSECount|PWIdth|RISe|RMS |SIGMA1|SIGMA2|SIGMA3|STDdev|WAVEFORMS}
MEASUrement:IMMed:TYPe?

Arguments AMPlitude measures the amplitude of the selected waveform. In other words, it measures the high value less the low value measured over the entire waveform or gated region. This measurement is applicable only to the analog channels.

\[ Amplitude = High - Low \]

AREa measures the voltage over time. The area is over the entire waveform or gated region and is measured in volt-seconds. The area measured above the ground is positive, while the area below ground is negative. This measurement is applicable only to the analog channels.
**BURST** measures the duration of a burst. The measurement is made over the entire waveform or gated region.

**CARea** (cycle area) measures the voltage over time. In other words, it measures, in volt-seconds, the area over the first cycle in the waveform or the first cycle in the gated region. The area measured above the common reference point is positive, while the area below the common reference point is negative. This measurement is applicable only to the analog channels.

**CMEan** (cycle mean) measures the arithmetic mean over the first cycle in the waveform or the first cycle in the gated region. This measurement is applicable only to the analog channels.

**CRMs** (cycle rms) measures the true Root Mean Square voltage over the first cycle in the waveform or the first cycle in the gated region. This measurement is applicable only to the analog channels.

**DELay** measures the time between the middle reference (default = 50%) amplitude point of the source waveform and the destination waveform.

**FALL** measures the time taken for the falling edge of the first pulse in the waveform or gated region to fall from a high reference value (default is 90%) to a low reference value (default is 10%). This measurement is applicable only to the analog channels.

**FREQuency** measures the first cycle in the waveform or gated region. Frequency is the reciprocal of the period and is measured in hertz (Hz), where 1 Hz = 1 cycle per second.

**HIGH** measures the High reference (100% level, sometimes called Topline) of a waveform. This measurement is applicable only to the analog channels.

**HITS** (histogram hits) measures the number of points in or on the histogram box.

**LOW** measures the Low reference (0% level, sometimes called Baseline) of a waveform. This measurement is applicable only to the analog channels.

**MAXimum** finds the maximum amplitude. This value is the most positive peak voltage found. It is measured over the entire waveform or gated region. This measurement is applicable only to the analog channels.

**MEAN** amplitude measurement finds the arithmetic mean over the entire waveform or gated region. This measurement is applicable only to the analog channels.

**MEDian** (histogram measurement) measures the middle point of the histogram box. Half of all acquired points within or on the histogram box are less than this value and half are greater than this value.

**MINimum** finds the minimum amplitude. This value is typically the most negative peak voltage. It is measured over the entire waveform or gated region. This measurement is applicable only to the analog channels.
NDuty (negative duty cycle) is the ratio of the negative pulse width to the signal period, expressed as a percentage. The duty cycle is measured on the first cycle in the waveform or gated region.

\[ \text{Negative Duty Cycle} = \left( \frac{\text{Negative Width}}{\text{Period}} \right) \times 100\% \]

NEDgeCount is the count of falling edges.

NOvershoot (negative overshoot) finds the negative overshoot value over the entire waveform or gated region. This measurement is applicable only to the analog channels.

\[ \text{Negative Overshoot} = \left( \frac{\text{Low} - \text{Minimum}}{\text{Amplitude}} \right) \times 100\% \]

NPulseCount is the count of negative pulses.

NWIdth (negative width) measurement is the distance (time) between the middle reference (default = 50%) amplitude points of a negative pulse. The measurement is made on the first pulse in the waveform or gated region.

PEAKhits measures the number of points in the largest bin of the histogram.

PDuty (positive duty cycle) is the ratio of the positive pulse width to the signal period, expressed as a percentage. It is measured on the first cycle in the waveform or gated region.

\[ \text{Positive Duty Cycle} = \left( \frac{\text{Positive Width}}{\text{Period}} \right) \times 100\% \]

PEDgeCount is the count of rising edges.

PERiod is the time required to complete the first cycle in a waveform or gated region. Period is the reciprocal of frequency and is measured in seconds.

PHASE measures the phase difference (amount of time a waveform leads or lags the reference waveform) between two waveforms. The measurement is made between the middle reference points of the two waveforms and is expressed in degrees, where 360° represents one waveform cycle.

PK2Pk (peak-to-peak) finds the absolute difference between the maximum and minimum amplitude in the entire waveform or gated region. This measurement is applicable only to the analog channels.

POvershoot is the positive overshoot value over the entire waveform or gated region. This measurement is applicable only to the analog channels.

\[ \text{Positive Overshoot} = \left( \frac{\text{Maximum} - \text{High}}{\text{Amplitude}} \right) \times 100\% \]

PPulseCount is the count of positive pulses.

PWIdth (positive width) is the distance (time) between the middle reference (default = 50%) amplitude points of a positive pulse. The measurement is made on the first pulse in the waveform or gated region.

RISE timing measurement finds the rise time of the waveform. The rise time is the time it takes for the leading edge of the first pulse encountered to rise from a
low reference value (default is 10%) to a high reference value (default is 90%).
This measurement is applicable only to the analog channels.

RMS amplitude measurement finds the true Root Mean Square voltage in the entire waveform or gated region. This measurement is applicable only to the analog channels.

SIGMA1 (histogram measurement) measures the percentage of points in the histogram that are within one standard deviation of the histogram mean.

SIGMA2 (histogram measurement) measures the percentage of points in the histogram that are within two standard deviations of the histogram mean.

SIGMA3 (histogram measurement) measures the percentage of points in the histogram that are within three standard deviations of the histogram mean.

STDdev measures the standard deviation (Root Mean Square (RMS) deviation) of all acquired points within or on the histogram box.

WAVEFORMS (waveform count) measures the number of waveforms used to calculate the histogram.

Examples

MEASUREMENT:IMMED:TYPE FREQUENCY defines the immediate measurement to be a frequency measurement.

MEASUREMENT:IMMED:TYPE? might return :MEASUREMENT:IMMED:TYPE RMS indicating that the immediate measurement is the true Root Mean Square voltage.

**MEASUrement:IMMed:UNIts? (Query Only)**

Returns a string containing the units of the immediate measurement.

**Group**
Measurement

**Syntax**
MEASUREMENT:IMMED:UNIts?

**Returns**
This query returns one of the following strings.

<table>
<thead>
<tr>
<th>Unit string</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;%&quot;</td>
<td>percent</td>
</tr>
<tr>
<td>&quot;/Hz&quot;</td>
<td>inverse hertz</td>
</tr>
<tr>
<td>&quot;p&quot;</td>
<td>indeterminate</td>
</tr>
<tr>
<td>&quot;A&quot;</td>
<td>amperes</td>
</tr>
<tr>
<td>&quot;A/A&quot;</td>
<td>amperes per ampere</td>
</tr>
<tr>
<td>&quot;A/V&quot;</td>
<td>amperes per volt</td>
</tr>
<tr>
<td>Unit string</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>“A/W”</td>
<td>ampere per watt</td>
</tr>
<tr>
<td>“A/dB”</td>
<td>amperes per decibel</td>
</tr>
<tr>
<td>“A/s”</td>
<td>amperes per second</td>
</tr>
<tr>
<td>“AA”</td>
<td>amperes squared</td>
</tr>
<tr>
<td>“AW”</td>
<td>ampere-watts</td>
</tr>
<tr>
<td>“AdB”</td>
<td>ampere-decibels</td>
</tr>
<tr>
<td>“As”</td>
<td>ampere-seconds</td>
</tr>
<tr>
<td>“B”</td>
<td>bytes</td>
</tr>
<tr>
<td>“Hz”</td>
<td>hertz</td>
</tr>
<tr>
<td>“IRE”</td>
<td>Institute of Radio Engineers units</td>
</tr>
<tr>
<td>“S/s”</td>
<td>samples per second</td>
</tr>
<tr>
<td>“V”</td>
<td>volts</td>
</tr>
<tr>
<td>“V/A”</td>
<td>volts per ampere</td>
</tr>
<tr>
<td>“V/V”</td>
<td>volts per volt</td>
</tr>
<tr>
<td>“V/W”</td>
<td>volts per watt</td>
</tr>
<tr>
<td>“VdB”</td>
<td>volts per decibel</td>
</tr>
<tr>
<td>“Vs”</td>
<td>volts per second</td>
</tr>
<tr>
<td>“VA”</td>
<td>volt-amperes</td>
</tr>
<tr>
<td>“VAR”</td>
<td>volt-amperes resistive</td>
</tr>
<tr>
<td>“VV”</td>
<td>volts squared</td>
</tr>
<tr>
<td>“VW”</td>
<td>volt-watts</td>
</tr>
<tr>
<td>“VdB”</td>
<td>volts-decibels</td>
</tr>
<tr>
<td>“Vs”</td>
<td>volt-seconds</td>
</tr>
<tr>
<td>“W”</td>
<td>watts</td>
</tr>
<tr>
<td>“W/A”</td>
<td>watts per ampere</td>
</tr>
<tr>
<td>“W/V”</td>
<td>watts per volt</td>
</tr>
<tr>
<td>“W/W”</td>
<td>watts per watt</td>
</tr>
<tr>
<td>“WdB”</td>
<td>watt per decibel</td>
</tr>
<tr>
<td>“Ws”</td>
<td>watts per second</td>
</tr>
<tr>
<td>“WA”</td>
<td>watt-amperes</td>
</tr>
<tr>
<td>“VV”</td>
<td>watt-volts</td>
</tr>
<tr>
<td>“WW”</td>
<td>watts squared</td>
</tr>
<tr>
<td>“WdB”</td>
<td>watt-decibels</td>
</tr>
<tr>
<td>“Wfms”</td>
<td>waveforms</td>
</tr>
<tr>
<td>“Ws”</td>
<td>watt-seconds</td>
</tr>
<tr>
<td>“dB”</td>
<td>decibels</td>
</tr>
<tr>
<td>“dB/A”</td>
<td>decibels per ampere</td>
</tr>
<tr>
<td>“dB/V”</td>
<td>decibels per volt</td>
</tr>
</tbody>
</table>
### Unit string Description

<table>
<thead>
<tr>
<th>Unit string</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;dB/W&quot;</td>
<td>decibels per watt</td>
</tr>
<tr>
<td>&quot;dB/dB&quot;</td>
<td>decibels per decibel</td>
</tr>
<tr>
<td>&quot;dBA&quot;</td>
<td>decibel-amperes</td>
</tr>
<tr>
<td>&quot;dBV&quot;</td>
<td>decibel-volts</td>
</tr>
<tr>
<td>&quot;dBW&quot;</td>
<td>decibel-watts</td>
</tr>
<tr>
<td>&quot;dBdB&quot;</td>
<td>decibels squared</td>
</tr>
<tr>
<td>&quot;day&quot;</td>
<td>days</td>
</tr>
<tr>
<td>&quot;degrees&quot;</td>
<td>degrees</td>
</tr>
<tr>
<td>&quot;div&quot;</td>
<td>divisions</td>
</tr>
<tr>
<td>&quot;edges&quot;</td>
<td>edges</td>
</tr>
<tr>
<td>&quot;hits&quot;</td>
<td>hits</td>
</tr>
<tr>
<td>&quot;hr&quot;</td>
<td>hours</td>
</tr>
<tr>
<td>&quot;joules&quot;</td>
<td>joules</td>
</tr>
<tr>
<td>&quot;min&quot;</td>
<td>minutes</td>
</tr>
<tr>
<td>&quot;ohms&quot;</td>
<td>ohms</td>
</tr>
<tr>
<td>&quot;percent&quot;</td>
<td>percent</td>
</tr>
<tr>
<td>&quot;pulses&quot;</td>
<td>pulses</td>
</tr>
<tr>
<td>&quot;s&quot;</td>
<td>seconds</td>
</tr>
<tr>
<td>&quot;unk&quot;</td>
<td>unknown</td>
</tr>
</tbody>
</table>

### Examples

MEASUREMENT:IMMED:UNITS? might return

:MEASUREMENT:IMMED:UNITS "s"

indicating that units for the immediate measurement are in seconds.

### MEASurement:IMMed:VALue? (Query Only)

Returns the value of the measurement specified by the MEASurement:IMMed:TYPe command. The measurement is immediately taken on the source(s) specified by a MEASurement:IMMed:SOURce<x> command.

**NOTE.** A change to HORIZontal:MAIn:SCALE or CH<x>:SCALE will not necessarily have taken affect if immediately followed by this command.

### Group

Measurement

### Syntax

MEASurement:IMMed:VALue?
Related Commands

MEASUrement:IMMed:TYPe, MEASUrement:IMMed:SOUrce<x>, *ESR?, ALLEv?

Examples

MEASUrement:IMMed:VALUE? might return :MEASUREMENT:IMMED:VALUE 9.9000E+37. If the measurement has an error or warning associated with it, then an item is added to the error queue. The error can be checked for with the *ESR? and ALLEv? commands.

MEASUrement:INDICators? (Query Only)

Returns all measurement indicator parameters.

Group
Measurement

Syntax
MEASUrement:INDICators?

Examples
MEASUREMENT:INDICATORS? might return
MEASUREMENT:INDICATORS:STATE MEAS1;NUMHORZ 0;NUMVERT 4;HORZ1 7.5E0;HORZ2 -3.400000095367E0;HORZ3 0.0E0;HORZ4 0.0E0;VERT1 -6.351123E-6;VERT2 -3.179753E-6;VERT3 -6.40943E-6;VERT4 -6.403E-6

MEASUrement:INDICators:HORZ<x>? (Query Only)

Returns the position of the specified horizontal measurement indicator <x>, where <x> can be 1, 2, 3, or 4.

Group
Measurement

Syntax
MEASUrement:INDICators:HORZ<x>?

Examples
MEASUREMENT:INDICATORS:HORZ1? might return
MEASUREMENT:INDICATORS:HORZ1 -2.0E-3 indicating that horizontal indicator1 has a value of -2mV.

MEASUrement:INDICators:NUMHORZ? (Query Only)

Returns the number of horizontal measurement indicators currently being displayed.
Commands Listed in Alphabetical Order

**Group**

**Measurement**

**Syntax**

MEASUrement:INDICators:NUMHORZ?

**Examples**

MEASUREMENT:INDICATORS:NUMHORZ? might return MEASUREMENT:INDICATORS:NUMHORZ 2 indicating there are currently 2 horizontal lines drawn on the graticule. The indicators show where the measurement specified by MEASUrement:INDICators:STATE is being performed.

**MEASUrement:INDICators:NUMVERT? (Query Only)**

Returns the number of vertical measurement indicators currently being displayed.

**Group**

**Measurement**

**Syntax**

MEASUrement:INDICators:NUMVERT?

**Examples**

MEASUREMENT:INDICATORS:NUMVERT? might return MEASUREMENT:INDICATORS:NUMVERT 2 indicating there are currently 2 vertical lines drawn on the graticule. The indicators show where the measurement specified by MEASUrement:INDICators:STATE is being performed.

**MEASUrement:INDICators:STATE**

Sets or returns the state of visible measurement indicators.

**Group**

**Measurement**

**Syntax**

MEASUrement:INDICators:STATE {OFF|MEAS<x>}

MEASUrement:INDICators:STATE?

**Arguments**

OFF turns the visible measurement indicators off.

MEAS<x> displays the visible measurement indicators for measurement <x>, where <x> can be 1 through 8.

**NOTE.** There must be an active measurement before you can activate an indicator for a specified measurement.
Examples  
MEASUREMENT:INDICATORS:STATE MEAS2 turns on the display of visible measurement indicators for measurement 2.

MEASUREMENT:INDICATORS:STATE? might return MEASUREMENT:INDICATORS:STATE OFF indicating that no measurement indicators are active.

MEASUREMENT:INDICATORS:VERT<x>? (Query Only)  
Returns the value of the specified vertical measurement indicator <x> from the trigger point, where <x> can be 1 through 8. A negative value means that the indicator is positioned earlier in the waveform record than the trigger point.

Group  Measurement  
Syntax  MEASUREMENT:INDICATORS:VERT<x>?

Examples  
MEASUREMENT:INDICATORS:VERT2? might return MEASUREMENT:INDICATORS:VERT2 -3.724507E-6 indicating that the second measurement indicator is positioned 3.72 μs before the trigger point.

MEASUREMENT:MEAS<x>? (Query Only)  
Returns all measurement parameters for the specified active measurement <x>.

Group  Measurement  
Syntax  MEASUREMENT:MEAS<x>?

MEASUREMENT:MEAS<x>:COUNt? (Query Only)  
Returns the number of values accumulated for this measurement since the last statistical reset. Values may be ignored if they generated an error. Measurements are specified by x, which ranges from 1 through 8.

Group  Measurement  
Syntax  MEASUREMENT:MEAS<x>:COUNt?
Examples


**MEASUrement:MEAS<x>:DELa y? (Query Only)**

Returns the delay measurement parameters for the measurement specified by <x>, which ranges from 1 through 8.

**Group**
Measurement

**Syntax**
MEASUrement:MEAS<x>:DELa y?

**Examples**
MEASUREMENT:MEAS1? might return :MEASUREMENT:MEAS1:DELa y:DIRECTION FORWARDS;EDGE1 RISE;EDGE2 RISE.

**MEASUrement:MEAS<x>:DELa y:DIRection**

Sets or returns the starting point and direction that determines the delay "to" edge when taking a delay measurement. Use the MEASUrement:MEAS<x>:SOUrce<x> command to specify the waveform.

**Group**
Measurement

**Syntax**
MEASUrement:MEAS<x>:DELa y:DIRection {BACKwards|FORw ards}
MEASUrement:MEAS<x>:DELa y:DIRection?

**Related Commands**
MEASUrement:MEAS<x>:SOUrce<x>

**Arguments**

BACKwards means the search starts at the end of the waveform and looks for the last rising or falling edge in the waveform. Use the MEASUrement:MEAS<x>:DELa y:EDGE<x> command to specify the slope of the edge.

FORwards means the search starts at the beginning of the waveform and looks for the first rising or falling edge in the waveform. Use the MEASUrement:MEAS<x>:DELa y:EDGE<x> command to specify the slope of the edge.
Examples
MEASUREMENT:MEAS3:DELAY:DIRECTION BACKWARDS starts searching from the end of the waveform record.


MEASUrement:MEAS<x>:DELay:EDGE<x>
Sets or returns the slope of the edge used for the delay "from" or "to" waveform when taking an immediate delay measurement. The waveform is specified by MEASUrement:MEAS<x>:SOUrce<x>.

Group
Measurement

Syntax
MEASUREMENT:MEAS<x>:DELAY:EDGE<x> {FALL|RISe}
MEASUREMENT:MEAS<x>:DELAY:EDGE<x>?

Arguments
<x> specifies which waveform to use, where <x> = 1 is the "from" waveform, and <x> = 2 is the "to" waveform.

FALL specifies the falling edge.

RISe specifies the rising edge.

Examples
MEASUREMENT:MEAS1:DELAY:EDGE1 RISE specifies that the "from" waveform rising edge be used for the immediate delay measurement.

MEASUREMENT:MEAS1:DELAY:EDGE1? returns either RISE or FALL.

MEASUrement:MEAS<x>:MAXimum? (Query Only)
Returns the maximum value found for this measurement since the last statistical reset. Measurements are specified by x, which ranges from 1 through 8.

Group
Measurement

Syntax
MEASUREMENT:MEAS<x>:MAXimum?

Examples
**MEASUrement:MEAS<x>:MEAN? (Query Only)**

Returns the mean value accumulated for this measurement since the last statistical reset. Measurements are specified by x, which ranges from 1 through 8.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:MEAN?


**MEASUrement:MEAS<x>:MINImum? (Query Only)**

Returns the minimum value for this measurement since the last statistical reset. Measurements are specified by <x>, which ranges from 1 through 8.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:MINImum?


**MEASUrement:MEAS<x>:SOUrce<x>**

For SOURce1: Sets or returns the source for all single channel measurements. For delay or phase measurements, sets or returns the waveform to measure "from".

For SOURce2: Sets or returns the waveform to measure "to" when taking a delay measurement or phase measurement.

Measurements are specified by <x>, which ranges from 1 to 8.

**NOTE.** If you do not specify a numeric suffix, the source is assumed to be SOURce1.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:SOUrce<x> {CH<x>|MATH|D<x>|HIStogram} MEASUrement:MEAS<x>:SOUrce<x>?
Commands Listed in Alphabetical Order

**Related Commands**

**Arguments**

CH<x> is an input channel waveform, where x is the channel number.

MATH is the math waveform.

REF<x> is a reference waveform, where x is the reference channel number.

D<x> is a digital waveform, where x is the digital channel number. Digital channels are available only on MSO models.

HISTogram indicates the histogram is the object to be measured. HISTogram only applies to SOURCE1; it is not allowed on SOURCE2.

**Examples**

MEASUREMENT:MEAS4:SOURCh1 specifies CH1 as the delay "to" source when making delay measurement.

MEASUREMENT:MEAS2:SOURCh2? might return MEASUREMENT:MEAS2:SOURCh2 MATH1 indicating that Math 1 is the measurement 2 source.

**MEASUREMENT:MEAS<x>:STATE**

Sets or returns whether the specified measurement slot is computed and displayed. The measurement slot is specified by <x>, which ranges from 1 through 8.

For a measurement to display, you must have selected a source waveform and defined the measurement you want to take and display. You select the measurement using the MEASUREMENT:MEAS<x>:SOURCh<x> command. You define the measurement type using the MEASUREMENT:MEAS<x>:TYPe command.

**Group**

Measurement

**Syntax**

```
MEASUREMENT:MEAS<x>:STATE {OFF|ON|<NR1>}
MEASUREMENT:MEAS<x>:STATE?
```

**Related Commands**

MEASUREMENT:MEAS<x>:SOURCh<x>, MEASUREMENT:MEAS<x>:TYPe

**Arguments**

OFF disables calculation and display of the specified measurement slot.

ON enables calculation and display of the specified measurement slot.

<NR1> = 0 disables calculation and display of the specified measurement slot; any other value enables calculation and display of the specified measurement slot.
**Examples**

MEASUREMENT:MEAS2:STATE ON computes and displays the measurement defined as measurement 2.

MEASUREMENT:MEAS1:STATE? might return :MEASUREMENT:MEAS1:STATE 0 indicating that measurement defined for measurement slot 1 is disabled.

**MEASUrement:MEAS<x>:STDdev? (Query Only)**

Returns the standard deviation of values accumulated for this measurement since the last statistical reset. Measurements are specified by <x>, the measurement slots, from 1 through 8.

**Group**
Measurement

**Syntax**
MEASUrement:MEAS<x>:STDdev?

**Examples**

**MEASUrement:MEAS<x>:TYPe**

Sets or returns the measurement type defined for the specified measurement slot. The measurement slot is specified by <x>, which ranges from 1 through 8.

Digital channel measurements do not have a user-settable midRef threshold. If you specify a digital channel measurement that is not available on MSO models, measurement error 2200: measurement system error occurs and 9.9e37 is returned.

**Group**
Measurement

**Syntax**
MEASUrement:MEAS<x>:TYPe {AMPlitude|AREa|BURst|CAREa|CMEan|CRMs|DElay|FALL|FREQuency|HIGH|HITS|LOW|MAXimum|MEAN|MEDian|MINimum|NDuty|NEDGECount|NOVersho|NPULSECount|NWIdth|PEAKHits|PDUty|PEDGECount|PERIod|PHase|PK2Pk|POVershoot|PPULSECount|PWIdth|RISe|RMS|SIGMA1|SIGMA2|SIGMA3|STDdev|WAVEFORMS}

**MEASUrement:MEAS<x>:TYPe?**

**Arguments**
AMPlitude measures the amplitude of the selected waveform. In other words, it measures the high value less the low value measured over the entire waveform or gated region. This measurement is applicable only to the analog channels.

Amplitude = High - Low
AREa measures the voltage over time. The area is over the entire waveform or gated region and is measured in volt-seconds. The area measured above the ground is positive, while the area below ground is negative. This measurement is applicable only to the analog channels.

BURst measures the duration of a burst. The measurement is made over the entire waveform or gated region.

CARea (cycle area) measures the voltage over time. In other words, it measures, in volt-seconds, the area over the first cycle in the waveform or the first cycle in the gated region. The area measured above the common reference point is positive, while the area below the common reference point is negative. This measurement is applicable only to the analog channels.

CMean (cycle mean) measures the arithmetic mean over the first cycle in the waveform or the first cycle in the gated region. This measurement is applicable only to the analog channels.

CRMs (cycle rms) measures the true Root Mean Square voltage over the first cycle in the waveform or the first cycle in the gated region. This measurement is applicable only to the analog channels.

DELay measures the time between the middle reference (default = 50%) amplitude point of the source waveform and the destination waveform. This measurement is applicable only to the analog channels.

FALL measures the time taken for the falling edge of the first pulse in the waveform or gated region to fall from a high reference value (default is 90%) to a low reference value (default is 10%). This measurement is applicable only to the analog channels.

FREQuency measures the first cycle in the waveform or gated region. Frequency is the reciprocal of the period and is measured in hertz (Hz), where 1 Hz = 1 cycle per second.

HIGH measures the High reference (100% level, sometimes called Topline) of a waveform. This measurement is applicable only to the analog channels.

HITS (histogram hits) measures the number of points in or on the histogram box.

LOW measures the Low reference (0% level, sometimes called Baseline) of a waveform. This measurement is applicable only to the analog channels.

MAXimum finds the maximum amplitude. This value is the most positive peak voltage found. It is measured over the entire waveform or gated region. This measurement is applicable only to the analog channels.

MEAN amplitude measurement finds the arithmetic mean over the entire waveform or gated region. This measurement is applicable only to the analog channels.

MEDian (histogram measurement) measures the middle point of the histogram box. Half of all acquired points within or on the histogram box are less than this value and half are greater than this value.
MINImum finds the minimum amplitude. This value is typically the most negative peak voltage. It is measured over the entire waveform or gated region. This measurement is applicable only to the analog channels.

NDuty (negative duty cycle) is the ratio of the negative pulse width to the signal period, expressed as a percentage. The duty cycle is measured on the first cycle in the waveform or gated region.

\[
\text{Negative Duty Cycle} = \left( \frac{\text{Negative Width}}{\text{Period}} \right) \times 100\%
\]

NEDGECount is the count of negative edges.

NOvershoot (negative overshoot) finds the negative overshoot value over the entire waveform or gated region. This measurement is applicable only to the analog channels.

\[
\text{Negative Overshoot} = \left( \frac{\text{Low} - \text{Minimum}}{\text{Amplitude}} \right) \times 100\%
\]

NPULSECount is the count of negative pulses.

NWIdth (negative width) measurement is the distance (time) between the middle reference (default = 50%) amplitude points of a negative pulse. The measurement is made on the first pulse in the waveform or gated region.

PEAKHİts measures the number of points in the largest bin of the histogram.

PDUty (positive duty cycle) is the ratio of the positive pulse width to the signal period, expressed as a percentage. It is measured on the first cycle in the waveform or gated region.

\[
\text{Positive Duty Cycle} = \left( \frac{\text{Positive Width}}{\text{Period}} \right) \times 100\%
\]

PEDGECount is the count of positive edges.

PERIod is the time required to complete the first cycle in a waveform or gated region. Period is the reciprocal of frequency and is measured in seconds.

PHase measures the phase difference (amount of time a waveform leads or lags the reference waveform) between two waveforms. The measurement is made between the middle reference points of the two waveforms and is expressed in degrees, where 360° represents one waveform cycle.

PK2Pk (peak-to-peak) finds the absolute difference between the maximum and minimum amplitude in the entire waveform or gated region. This measurement is applicable only to the analog channels.

POvershoot is the positive overshoot value over the entire waveform or gated region. This measurement is applicable only to the analog channels.

\[
\text{Positive Overshoot} = \left( \frac{\text{Maximum} - \text{High}}{\text{Amplitude}} \right) \times 100\%
\]

PPULSECount is the count of positive pulses.
PWIdth (positive width) is the distance (time) between the middle reference (default = 50%) amplitude points of a positive pulse. The measurement is made on the first pulse in the waveform or gated region.

RISe timing measurement finds the rise time of the waveform. The rise time is the time it takes for the leading edge of the first pulse encountered to rise from a low reference value (default is 10%) to a high reference value (default is 90%). This measurement is applicable only to the analog channels.

RMS amplitude measurement finds the true Root Mean Square voltage in the entire waveform or gated region. This measurement is applicable only to the analog channels.

SIGMA1 (histogram measurement) measures the percentage of points in the histogram that are within one standard deviation of the histogram mean.

SIGMA2 (histogram measurement) measures the percentage of points in the histogram that are within two standard deviations of the histogram mean.

SIGMA3 (histogram measurement) measures the percentage of points in the histogram that are within three standard deviations of the histogram mean.

STDdev measures the standard deviation (Root Mean Square (RMS) deviation) of all acquired points within or on the histogram box.

WAVEFORMS (waveform count) measures the number of waveforms used to calculate the histogram.

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.

MEASUREmenteMEAS<x>:UNItS? (Query Only)

Returns the units associated with the specified measurement. The measurement slots are specified by <x>, which ranges from 1 through 8.
<table>
<thead>
<tr>
<th>Unit string</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;%&quot;</td>
<td>percent</td>
</tr>
<tr>
<td>&quot;/Hz&quot;</td>
<td>inverse hertz</td>
</tr>
<tr>
<td>&quot;?&quot;</td>
<td>indeterminate</td>
</tr>
<tr>
<td>&quot;A&quot;</td>
<td>amperes</td>
</tr>
<tr>
<td>&quot;A/A&quot;</td>
<td>amperes per ampere</td>
</tr>
<tr>
<td>&quot;A/V&quot;</td>
<td>amperes per volt</td>
</tr>
<tr>
<td>&quot;A/W&quot;</td>
<td>ampere per watt</td>
</tr>
<tr>
<td>&quot;A/dB&quot;</td>
<td>amperes per decibel</td>
</tr>
<tr>
<td>&quot;A/s&quot;</td>
<td>amperes per second</td>
</tr>
<tr>
<td>&quot;AA&quot;</td>
<td>amperes squared</td>
</tr>
<tr>
<td>&quot;AW&quot;</td>
<td>ampere-watts</td>
</tr>
<tr>
<td>&quot;AdB&quot;</td>
<td>ampere-decibels</td>
</tr>
<tr>
<td>&quot;As&quot;</td>
<td>ampere-seconds</td>
</tr>
<tr>
<td>&quot;B&quot;</td>
<td>bytes</td>
</tr>
<tr>
<td>&quot;Hz&quot;</td>
<td>hertz</td>
</tr>
<tr>
<td>&quot;IRE&quot;</td>
<td>Institute of Radio Engineers units</td>
</tr>
<tr>
<td>&quot;S/s&quot;</td>
<td>samples per second</td>
</tr>
<tr>
<td>&quot;V&quot;</td>
<td>volts</td>
</tr>
<tr>
<td>&quot;V/A&quot;</td>
<td>volts per ampere</td>
</tr>
<tr>
<td>&quot;V/V&quot;</td>
<td>volts per volt</td>
</tr>
<tr>
<td>&quot;V/W&quot;</td>
<td>volts per watt</td>
</tr>
<tr>
<td>&quot;V/dB&quot;</td>
<td>volts per decibel</td>
</tr>
<tr>
<td>&quot;V/s&quot;</td>
<td>volts per second</td>
</tr>
<tr>
<td>&quot;VA&quot;</td>
<td>volt-amperes</td>
</tr>
<tr>
<td>&quot;VAR&quot;</td>
<td>volt-amperes resistive</td>
</tr>
<tr>
<td>&quot;V/N&quot;</td>
<td>volts squared</td>
</tr>
<tr>
<td>&quot;VW&quot;</td>
<td>volt-watts</td>
</tr>
<tr>
<td>&quot;VdB&quot;</td>
<td>volts-decibels</td>
</tr>
<tr>
<td>&quot;Vs&quot;</td>
<td>volt-seconds</td>
</tr>
<tr>
<td>&quot;W&quot;</td>
<td>watts</td>
</tr>
<tr>
<td>&quot;W/A&quot;</td>
<td>watts per ampere</td>
</tr>
<tr>
<td>&quot;W/V&quot;</td>
<td>watts per volt</td>
</tr>
<tr>
<td>&quot;W/W&quot;</td>
<td>watts per watt</td>
</tr>
<tr>
<td>&quot;W/dB&quot;</td>
<td>watt per decibel</td>
</tr>
<tr>
<td>&quot;W/s&quot;</td>
<td>watts per second</td>
</tr>
<tr>
<td>&quot;WA&quot;</td>
<td>watt-amperes</td>
</tr>
<tr>
<td>&quot;WV&quot;</td>
<td>watt-volts</td>
</tr>
<tr>
<td>&quot;WW&quot;</td>
<td>watts squared</td>
</tr>
</tbody>
</table>
### Unit string

<table>
<thead>
<tr>
<th>Unit string</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;WdB&quot;</td>
<td>watt-decibels</td>
</tr>
<tr>
<td>&quot;Wfms&quot;</td>
<td>waveforms</td>
</tr>
<tr>
<td>&quot;Ws&quot;</td>
<td>watt-seconds</td>
</tr>
<tr>
<td>&quot;dB&quot;</td>
<td>decibels</td>
</tr>
<tr>
<td>&quot;dB/A&quot;</td>
<td>decibels per ampere</td>
</tr>
<tr>
<td>&quot;dB/V&quot;</td>
<td>decibels per volt</td>
</tr>
<tr>
<td>&quot;dB/W&quot;</td>
<td>decibels per watt</td>
</tr>
<tr>
<td>&quot;dB/dB&quot;</td>
<td>decibels per decibel</td>
</tr>
<tr>
<td>&quot;dBA&quot;</td>
<td>decibel-amperes</td>
</tr>
<tr>
<td>&quot;dBV&quot;</td>
<td>decibel-volts</td>
</tr>
<tr>
<td>&quot;dBW&quot;</td>
<td>decibel-watts</td>
</tr>
<tr>
<td>&quot;dBdB&quot;</td>
<td>decibels squared</td>
</tr>
<tr>
<td>&quot;day&quot;</td>
<td>days</td>
</tr>
<tr>
<td>&quot;degrees&quot;</td>
<td>degrees</td>
</tr>
<tr>
<td>&quot;div&quot;</td>
<td>divisions</td>
</tr>
<tr>
<td>&quot;edges&quot;</td>
<td>edges</td>
</tr>
<tr>
<td>&quot;hits&quot;</td>
<td>hits</td>
</tr>
<tr>
<td>&quot;hr&quot;</td>
<td>hours</td>
</tr>
<tr>
<td>&quot;joules&quot;</td>
<td>joules</td>
</tr>
<tr>
<td>&quot;min&quot;</td>
<td>minutes</td>
</tr>
<tr>
<td>&quot;ohms&quot;</td>
<td>ohms</td>
</tr>
<tr>
<td>&quot;percent&quot;</td>
<td>percent</td>
</tr>
<tr>
<td>&quot;pulses&quot;</td>
<td>pulses</td>
</tr>
<tr>
<td>&quot;s&quot;</td>
<td>seconds</td>
</tr>
<tr>
<td>&quot;unk&quot;</td>
<td>unknown</td>
</tr>
</tbody>
</table>

### Examples

MEASUREMENT:MEAS1:UNITS? might return :MEASUREMENT:MEAS1:UNIts % indicating units for measurement 1 are set to percent.

### MEASUrement:MEAS<x>:VALue? (Query Only)

Returns a calculate value for the measurement specified by <x>, which ranges from 1 through 8.

**NOTE.** This is the same value as displayed on-screen. If measurement statistics are enabled, a new value is calculated with every waveform. In addition, this value is updated approximately every 1/3 second. If you are acquiring a long acquisition record, the oscilloscope may take longer to update.
**Commands Listed in Alphabetical Order**

**Group**  
Measurement

**Syntax**  
MEASUrement:MEAS<x>:VALue?

**Related Commands**  
MEASUrement:MEAS<x>:UNIts?, *ESR?, ALLEv?

**Examples**  
MEASUREMENT:MEAS1:VALUE? might return :MEASUREMENT:MEAS1:VALue 2.8740E-06. If the measurement has an error or warning associated with it, then an item is added to the error queue. The error can be checked for with the *ESR? and ALLEv? commands.

**MEASUrement:METHod**  
Sets or returns the method used to calculate the 0% and 100% reference level.

**Group**  
Measurement

**Syntax**  
MEASUrement:METHod {Auto|HIStogram|MINMax}  
MEASUrement:METHod?

**Related Commands**  

**Arguments**  
Auto selects the best method for each data set.  
HIStogram sets the high and low waveform levels statistically using a histogram algorithm.  
MINMax uses the highest and lowest values of the waveform record. This selection is best for examining waveforms with no large, flat portions of a common value, such as sine waves and triangle waves.

**Examples**  
MEASUREMENT:METHod? might return :MEASUREMENT:METHOD MINMAX indicating that the reference levels are set to MIN and MAX.

**MEASUrement:REFLevel? (Query Only)**  
Returns the current reference level parameters.
Group       Measurement

Syntax      MEASurement:REFLevel?

Examples    MEASUREMENT:REFLEVEL? might return these reference level settings:
            :MEASUREMENT:REFLEVEL:METHOD PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2 0.0000;:MEASUREMENT:REFLEVEL:PERCENT:HIGH 90.0000;LOW 10.0000;MID1 50.0000;MID2 50.0000

**MEASUrement:REFLevel:ABSolute:HIGH**

Sets or returns the high reference level, and is the upper reference level when MEASUrement:REFLevel:METHod is set to Absolute. This command affects the results of rise and fall measurements.

*NOTE.* This command affects the associated reference level parameter for all MEASurements:IMMed and the four periodic measurements.

Group       Measurement

Syntax      MEASUrement:REFLeveL:ABSolute:HIGH <NR3>
               MEASUrement:REFLevel:ABSolute:HIGH?

Related Commands

MEASUrement:REFLevel:METHod, MEASUrement:IMMed:TYPe, MEASUrement:MEAS<x>:TYPe

Arguments   <NR3> is the high reference level, in volts. The default is 0.0 V.

Examples    MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH 1.71 sets the high reference level to 1.71 V.
            MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH? might return :MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH 1.7100E+00 indicating that the absolute high reference level is set to 1.71 V.

**MEASUrement:REFLevel:ABSolute:LOW**

Sets or returns the low reference level, and is the lower reference level when MEASUrement:REFLevel:METHod is set to Absolute.
Commands Listed in Alphabetical Order

### MEASUrement:REFLevel:ABSolute:LOW

**NOTE.** This command affects the associated reference level parameter for all MEASurements:IMMed and the four periodic measurements.

<table>
<thead>
<tr>
<th>Group</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related Commands</td>
<td>MEASUrement:REFLevel:METHod, MEASUrement:IMMed:TYPe, MEASUrement:MEAS&lt;x&gt;:TYPe</td>
</tr>
<tr>
<td>Arguments</td>
<td>&lt;NR3&gt; is the low reference level, in volts. The default is 0.0 V.</td>
</tr>
<tr>
<td>Examples</td>
<td>MEASUREMENT:REFLEVEL:ABSOLUTE:LOW 0.0 sets the low reference level to 0.0 V. MEASUREMENT:REFLEVEL:ABSOLUTE:LOW? might return :MEASUREMENT:REFLEVEL:ABSOLUTE:LOW 0.0000E+00 indicating that the absolute low reference level is set to 0.0 V.</td>
</tr>
</tbody>
</table>

### MEASUrement:REFLevel:ABSolute:MID[1]
Sets or returns the mid reference level, and is the 50% reference level when MEASUrement:REFLevel:METHod is set to Absolute. This command affects the results of period, frequency, delay, and all cyclic measurements.

**NOTE.** This command affects the associated reference level parameter for all MEASurements:IMMed and the four periodic measurements.

<table>
<thead>
<tr>
<th>Group</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related Commands</td>
<td>MEASUrement:REFLevel:METHod</td>
</tr>
<tr>
<td>Arguments</td>
<td>&lt;NR3&gt; is the mid reference level, in volts. The default is 0.0 V.</td>
</tr>
</tbody>
</table>
**Examples**

MEASUREMENT:REFLEVEL:ABSOLUTE:MID 1 .71 sets the mid reference level to .71 V.

MEASUREMENT:REFLEVEL:ABSOLUTE:MID? might return:
:MEASUREMENT:REFLEVEL:ABSOLUTE:MID 0.7100E+00 indicating that the absolute mid1 reference level is set to .71 V.

**MEASUrement:REFLevel:ABSolute:MID2**

Sets or returns the mid reference level for the "to" waveform when taking a delay measurement, and is the 50% reference level when MEASUREMENT:REFLevel:METHod is set to Absolute. This command affects the results of delay measurements.

**NOTE.** this command affects the associated reference level parameter for all MEASurements:IMMed and the four periodic measurements.

**Group**
Measurement

**Syntax**

MEASUrement:REFLevel:ABSolute:MID2 <NR3>

MEASUrement:REFLevel:ABSolute:MID2?

**Related Commands**
MEASUrement:REFLevel:METHod

**Arguments**

<NR3> is the mid reference level, in volts. The default is 0.0 V.

**Examples**

MEASUREMENT:REFLEVEL:ABSOLUTE:MID2 0.5 sets the mid reference level for the delay waveform to 0.5 V.

MEASUREMENT:REFLEVEL:ABSOLUTE:MID2? might return:
:MEASUREMENT:REFLEVEL:ABSOLUTE:MID2 0.5000E+00 indicating that the absolute mid2 reference level is set to 0.5 V.

**MEASUrement:REFLevel:ABSolute:MID<x>**

Sets or returns the mid reference level for channel <x>, where x is the measurement channel.

**Group**
Measurement
**Syntax**

```
MEASurement:REFLevel:ABSolute:MID<x> <NR3>
MEASurement:REFLevel:ABSolute:MID<x>?
```

**Arguments**

<NR3> is the mid reference level in volts.

---

**MEASUrement:REFLevel:METHod**

Specifies or returns the reference level units used for measurement calculations.

**NOTE.** This command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.

---

**Group** Measurement

**Syntax**

```
MEASurement:REFLevel:METHod {ABSolute|PERCent}
MEASurement:REFLevel:METHod?
```

**Arguments**

- **ABSolute** specifies that the reference levels are set explicitly using the MEASUrement:REFLevel:ABSolute commands. This method is useful when precise values are required (for example, when designing to published interface specifications, such as RS-232-C).

- **PERCent** specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUrement:REFLevel:PERCent commands.

**Examples**

```
MEASUREMENT:REFLEVEL:METHOD ABSOLUTE specifies that explicit user-defined values are used for the reference levels.

MEASUREMENT:REFLEVEL:METHOD? might return :MEASUREMENT:REFLEVEL:METHOD PERCENT indicating that the reference level units used are calculated as a percent relative to HIGH and LOW.
```

---

**MEASUrement:REFLevel:PERCent:HIGH**

Sets or returns the percent (where 100% is equal to HIGH) used to calculate the high reference level when MEASUrement:REFLevel:METHod is set to Percent. This command affects the results of rise and fall measurements.
### MEASurement:REFLevel:PERCent:HIGH

**NOTE.** This command affects the associated reference level parameter for all MEASurements:IMMed and the four periodic measurements.

<table>
<thead>
<tr>
<th>Group</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td><code>MEASurement:REFLevel:PERCent:HIGH &lt;NR3&gt;</code>&lt;br&gt;<code>MEASurement:REFLevel:PERCent:HIGH?</code></td>
</tr>
<tr>
<td>Related Commands</td>
<td><code>MEASurement:REFLevel:METHod, MEASurement:IMMed:TYPe, MEASurement:MEAS&lt;x&gt;:TYPe</code></td>
</tr>
<tr>
<td>Arguments</td>
<td><code>&lt;NR3&gt;</code> is the high reference level, ranging from 0 to 100%. The default high reference level is 90%.</td>
</tr>
<tr>
<td>Examples</td>
<td><code>MEASUREMENT:REFLEVEL:PERCENT:HIGH 95</code> sets the high reference level to 95% of HIGH.&lt;br&gt;<code>MEASUREMENT:REFLEVEL:PERCENT:HIGH?</code> might return <code>:MEASUREMENT:REFLEVEL:PERCENT:HIGH 90</code> indicating that the percentage high reference level is set to 90% of HIGH.</td>
</tr>
</tbody>
</table>

### MEASurement:REFLevel:PERCent:LOW

Sets or returns the percent (where 100% is equal to HIGH) used to calculate the low reference level when MEASurement:REFLevel:METHod is set to Percent. This command affects the results of rise and fall measurements.

**NOTE.** This command affects the associated reference level parameter for all MEASurements:IMMed and the four periodic measurements.

<table>
<thead>
<tr>
<th>Group</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td><code>MEASurement:REFLevel:PERCent:LOW &lt;NR3&gt;</code>&lt;br&gt;<code>MEASurement:REFLevel:PERCent:LOW?</code></td>
</tr>
<tr>
<td>Related Commands</td>
<td><code>MEASurement:REFLevel:METHod, MEASurement:IMMed:TYPe, MEASurement:MEAS&lt;x&gt;:TYPe</code></td>
</tr>
</tbody>
</table>
Arguments

<NR3> is the low reference level, ranging from 0 to 100%. The default low reference level is 10%.

Examples

MEASUREMENT:REFLEVEL:PERCENT:LOW 15 sets the high reference level to 15% of HIGH.

MEASUREMENT:REFLEVEL:PERCENT:LOW? might return :MEASUREMENT:REFLEVEL:PERCENT:LOW 10 indicating that the percentage high reference level is set to 10% of HIGH.

MEASUREMENT:REFLevel:PERCent:MID[1]

Sets or returns the percent (where 100% is equal to HIGH) that is used to calculate the mid reference level when MEASUrement:REFLevel:METHod is set to Percent. This command affects the results of period, frequency, delay, and all cyclic measurements.

NOTE. this command affects the associated reference level parameter for all MEASurements:IMMed and the four periodic measurements.

Group

Measurement

Syntax


MEASUrement:REFLevel:PERCent:MID[1]?

Related Commands

MEASUrement:REFLevel:METHod

Arguments

<NR3> is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

Examples

MEASUREMENT:REFLEVEL:PERCENT:MID 1 60 sets the mid reference level to 60% of HIGH.

MEASUREMENT:REFLEVEL:PERCENT:MID? might return :MEASUREMENT:REFLEVEL:PERCENT:MID 65 indicating that the percentage mid reference level is set to 65% of HIGH.

MEASUrement:REFLevel:PERCent:MID2

Sets or returns the percent (where 100% is equal to HIGH) that is used to calculate the mid reference level for the second waveform specified when
**MEASUrement:REFLevel:METHod** is set to Percent. This command affects the results of delay measurements.

**NOTE.** This command affects the associated reference level parameter for all **MEASUrement:IMMed** and the four periodic measurements.

<table>
<thead>
<tr>
<th>Group</th>
<th>Measurement</th>
</tr>
</thead>
</table>
| Syntax    | **MEASUrement:REFLevel:PERCent:MID2** <NR3>  
**MEASUrement:REFLevel:PERCent:MID2?** |
| Related Commands | **MEASUrement:REFLevel:METHod** |
| Arguments | <NR3> is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%. |
| Examples  | **MEASUREMENT:REFLEVEL:PERCENT:MID2 40** sets the mid2 reference level to 40% of HIGH.  
**MEASUREMENT:REFLEVEL:PERCENT:MID2?** might return :**MEASUREMENT:REFLEVEL:PERCENT:MID2 45** indicating that the percentage mid2 reference level is set to 45% of HIGH. |

**MEASUrement:REFLevel:PERCent:MID<x>**

Sets or returns the mid reference level for channel <x>, where x is the measurement channel.

<table>
<thead>
<tr>
<th>Group</th>
<th>Measurement</th>
</tr>
</thead>
</table>
| Syntax    | **MEASUrement:REFLevel:PERCent:MID<x>** <NR3>  
**MEASUrement:REFLevel:PERCent:MID<x>?** |
| Arguments | <NR3> is the mid reference level in percent. |

**MEASUrement:SNAPShot (No Query Form)**

Displays the measurement snapshot list on the oscilloscope screen. The list contains the immediate values for all available measurements of the active signal.
Commands Listed in Alphabetical Order

**Group** Measurement

**Syntax** MEASurement:SNAPShot

**MEASUrement:STATIstics (No Query Form)**

Clears all of the statistics accumulated for all periodic measurements (MEAS1 through MEAS4).

The query form returns statistic settings.

**Group** Measurement

**Syntax** MEASurement:STATIstics RESET

MEASurement:STATIstics?

**Arguments** RESET clears the measurements.

**MEASUrement:STATIstics:MODE**

Controls the operation and display of management statistics.

**Group** Measurement

**Syntax** MEASUrement:STATIstics:MODE {OFF|ON}

MEASUrement:STATIstics:MODE?

**Related Commands** MEASUrement:STATIstics

**Arguments** OFF turns all measurements off. This is the default value.

ON turns on statistics and displays all statistics for each measurement.

**Examples** MEASUREMENT:STATISTICS:MODE OFF turns statistic measurements off.

MEASUREMENT:STATISTICS:MODE? might return

:MEASUREMENT:STATISTICS:MODE ON indicating that measurement statistics are turned on and all statistics are being displayed for each measurement.
MEASUrement:STATIstics:WEIghting

Sets or returns the time constant for mean and standard deviation statistical accumulations.

**Group**  
Measurement

**Syntax**  
MEASUrement:STATIstics:WEIghting <NR1>  
MEASUrement:STATIstics:WEIghting?

**Related Commands**  
MEASUrement:STATIstics:MODE

**Arguments**  
<NR1> is the number of samples used for the mean and standard deviation statistical accumulations.

**Examples**  
MEASUREMENT:STATISTICS:WEIGHTING 4 sets statistical weighting to four samples.  
MEASUREMENT:STATISTICS:WEIGHTING? might return :MEASUREMENT:STATISTICS:WEIGHTING 4 indicating that measurement statistics weighting is currently set to 4 samples.

MESSage

This command sets or queries message box (screen annotation) parameters.

**Group**  
Display

**Syntax**  
MESSage  
MESSage?

**Examples**  
MESSAGE? might return MESSAGE:SHOW "TP401";BOX 271,82,292,114;STATE 0 indicating the message parameters.

MESSage:BOX

Sets or returns the co-ordinates of the message box. This command does not display the message unless MESSage:STATE is on.
X1 and Y1 are the screen coordinates of the top left corner of the message box. X2 and Y2 are the screen coordinates of the bottom right corner of the message box. All four coordinates are returned by the query.

Changing the text in the message box, using the MESSAGE:SHOW command, automatically resizes the message box. If you want a custom message box size, send the MESSAGE:BOX command after changing the text using the MESSAGE:SHOW command.

Message box settings and data are saved and restored in saved setups.

**Group** Display

**Syntax**

```
MESSage:BOX <X1>,<Y1>[,<X2>,<Y2>]
MESSage:BOX?
```

**Related Commands** MESSage:STATE, MESSage:SHOW, MESSage:CLEAR

**Arguments**

- `<X1>` and `<X2>` = 0 to 1023, and are pixel positions along the horizontal axis. `<X1>` defines the left and `<X2>` defines the right side of the window.
- `<Y1>` and `<Y2>` = 0 to 767, and are pixel positions along the vertical axis. `<Y1>` defines the top and `<Y2>` defines the bottom of the window. The reserved height of all characters is 16 pixels so the window must be at least that high to fully display characters. `<X2>` and `<Y2>` are optional because the MESSAGE:SHOW command automatically sizes the box to fit the message. All four values are returned in a query.

**MESSage:CLEAR (No Query Form)**

Clears the contents of the message box.

**Group** Display

**Syntax** MESSage:CLEAR

**Related Commands** MESSage:BOX, MESSage:SHOW, MESSage:STATE

**Examples** MESSage:CLEAR removes the message from the message window.
MESSage:SHOW

Sets or returns the contents of the message box. MESSage:SHOW <Qstring> defines the content of the message box. Change in string length causes automatic resize of the message box to fit the text. The box may be resized using the MESSage:BOX command. The MESSage:STATE command is used to turn on and off the message box display.

**Group**  
Display

**Syntax**

MESSage:SHOW <Qstring>  
MESSage:SHOW?

**Related Commands**

MESSage:BOX, MESSage:CLEAR, MESSage:STATE

**Arguments**

<Qstring> is the message and can include any of the characters shown in the Character Set, Appendix A. The maximum length of the message is 1000 characters; the instrument ignores longer strings.

The message box size is set to fit the message. You can also set the message area height and width using the MESSage:BOX command. The length of the message that fits in the message area depends on the contents of the message because the width of characters varies.

If the message exceeds the limits of the message box, either horizontally or vertically, the portion of the message that exceeds the limits will not be displayed. The message string itself is not altered. The entire message can be returned as a query response regardless of what is displayed in the message box.

The message is left-justified, and is displayed on a single line starting with the top most line in the window. A new line character can be embedded in the string to position the message on multiple lines. You can also use white space and tab characters to position the message within a line. Text which does not fit within the message box is truncated. Defining a message box text string erases any previously displayed text within the message box.

You can send a tab by transmitting a tab character (\t or \x09) followed characters representing the most significant eight bits followed by significant eight bits of a 16-bit number. The number specifies the position relative to the left margin of the message area. For example, to tab send TAB (\t or \x09), NUL (decimal 0), and CR (decimal 13).

For example, using hexadecimal escape sequences, MESSage:SHOW ’\x09\x01\x17Hello’ when sent as a command would cause the ’Hello’ to be displayed starting at pixel position 279 relative to the left margin set by the MESSage:BOX command. If you want to display characters starting at position
279, then \(279 = 0x0117\); split the hexadecimal number into two characters 0x01 and 0x17 and send \x09\x01\x17.

Special characters which control decoration are two character sequences where the first character is an escape (0x1b) and the second character is as described below.

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 6</td>
<td>If set, inverse video is toggled from current state and the following text is displayed in the new inverse state until the state is toggled again. Remaining bits are ignored</td>
</tr>
<tr>
<td>Bit 5</td>
<td>If set, the color index in the four LSB's (bits 0 through 3) is applied to the foreground or background color depending on the fg/bg bit (bit 4).</td>
</tr>
<tr>
<td>Bit 4</td>
<td>If set, color change is applied to the background, otherwise applies to the foreground.</td>
</tr>
<tr>
<td>Bit 0 – 3</td>
<td>Specifies the color index (0 through 15) to change color as specified below:</td>
</tr>
<tr>
<td>Index 0</td>
<td>Black (background)</td>
</tr>
<tr>
<td>Index 1</td>
<td>Yellow (Ch 1)</td>
</tr>
<tr>
<td>Index 2</td>
<td>Cyan (Ch 2)</td>
</tr>
<tr>
<td>Index 3</td>
<td>Magenta (Ch 3)</td>
</tr>
<tr>
<td>Index 4</td>
<td>Green (Ch 4)</td>
</tr>
<tr>
<td>Index 5</td>
<td>Red (math)</td>
</tr>
<tr>
<td>Index 6</td>
<td>White (reference)</td>
</tr>
<tr>
<td>Index 7</td>
<td>Orange</td>
</tr>
<tr>
<td>Index 8</td>
<td>Gray (Graticule)</td>
</tr>
<tr>
<td>Index 9</td>
<td>White (text)</td>
</tr>
<tr>
<td>Index 10</td>
<td>Tek blue</td>
</tr>
<tr>
<td>Index 11</td>
<td>Bright blue</td>
</tr>
<tr>
<td>Index 12</td>
<td>Undefined</td>
</tr>
<tr>
<td>Index 13</td>
<td>Blue</td>
</tr>
<tr>
<td>Index 14</td>
<td>Undefined</td>
</tr>
<tr>
<td>Index 15</td>
<td>Dark blue</td>
</tr>
<tr>
<td>Bit 4</td>
<td>If set, the foreground color is set to the default foreground color.</td>
</tr>
<tr>
<td>Bit 3</td>
<td>If set, the background color is set to the default background color.</td>
</tr>
<tr>
<td>Bit 2</td>
<td>Undefined</td>
</tr>
<tr>
<td>Bit 1</td>
<td>Undefined</td>
</tr>
<tr>
<td>Bit 0</td>
<td>Undefined</td>
</tr>
</tbody>
</table>

The ESC (escape) character followed by the @ character turns inverse video on or off and can be embedded in the message string. Example: “abcdefESC@defESC@ghi” specifies the string “abcdefghi” where the “def” portion is displayed in inverse video.
Example: “abcESC#defESC)ESC@ghi” specifies the string “abedefghi” where the “def” portion appears in the channel 3 color (magenta) and the “ghi” portion appears in the normal text color except it’s in inverse video.

An alternate way to enter characters is octal escape sequences. This consists of a backslash followed by numerals in the standard C language printf fashion.

Another way to enter characters is \nnn where the nn is the hexadecimal value of the character to display.

An advantage of these methods is that any controller program can be used. Another advantage is it’s easy to access characters with the high bit set, that is, those characters with a decimal value greater than 127.

An alternate way to enter certain characters is with a backslash followed by a single character (following “standard” Unix) as described in the table below.

<table>
<thead>
<tr>
<th>n</th>
<th>Newline (carriage return and line feed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>\</td>
<td>Backslash (\ is required to get a backslash character)</td>
</tr>
<tr>
<td>t</td>
<td>Horizontal tab; the next 2 characters specify the pixel column to tab to as explained earlier</td>
</tr>
</tbody>
</table>

If a backslash is followed by an undefined special character, the backslash is ignored and the character following it is accepted as is.

NOTE. The use of any escape codes other than those described above may produce unpredictable results.

Examples

MESSage:SHOW “Hello World”
displays “Hello world” in the upper left corner of the box (you can define the box size with the MESSAGE BOX command).

MESSage:SHOW “↵Hello World↵... hello”
displays “Hello world ... hello” in the upper left corner of the box and the word “world” is displayed in inverse video. In this example, ↵ stands for the escape character. The escape character may appear differently for you depending on your controller program.

MESSage:STATE

Controls the display of the message box.

Group
Display

Syntax
MESSage:STATE {OFF|ON|<NR1>}
MESSage:STATE?
Related Commands

MESSage:BOX
MESSage:SHOW, MESSage:CLEAR

Arguments

OFF or <NR1> = 0 removes the message window from the screen.
ON or <NR1> ≠ 0 displays the message window and its contents on the screen.

NEWpass (No Query Form)

This command 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
*PUD
PASSWord

Arguments
<QString> is the new password, which can contain up to 16 characters.

Examples
NEWPASS "mypassword" creates a new password (mypassword) for accessing your protected data.

*OPC

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. (See page 3-1, Status and Events.)

The *OPC command allows you to synchronize the operation of the oscilloscope with your application program. (See page 3-7, Synchronization Methods.)

Certain oscilloscope operations can affect the *OPC response. (See Table 3-3 on page 3-7.)

Group
Status and Error
**Commands Listed in Alphabetical Order**

### *OPC

**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.

### PASSWORD(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 oscilloscope is powered off, or until the FACTory command or the PASSWORD command with no arguments 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 and can include 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 *PUB and NEWpass set commands. You can still use the query version of *PUD.

### PICTBridge:DATEPrint

Enables or disables printing the date on the print output

**Group**

PictBridge
**PICTBridge:DATEPrint**

**Syntax**

```
PICTBridge:DATEPrint {DEFLT|OFF|ON}
```

**Arguments**

- **DEFLT** is the default setting of the printer.
- **ON** enables the date print on the print output.
- **OFF** disables the date print on the print output.

**Examples**

- `PICTBRIDGE:DATEPRINT ON` enables the date print on the print output.
- `PICTBRIDGE:DATEPRINT?` might return `PICTBRIDGE:DATEPRINT DEFLT` indicating the default setting of the printer.

---

**PICTBridge:DEFault (No Query Form)**

Sets the arguments for all PictBridge commands to their default values. The default values are same as printer default settings.

**Group**

PictBridge

**Syntax**

```
PICTBridge:DEFault
```

**Examples**

- `PICTBRIDGE:DEFAULT` sets all PICTBridge commands to their default values.

---

**PICTBridge:IDPrint**

Enables or disables printing the oscilloscope model and serial number on the print output.

**Group**

PictBridge

**Syntax**

```
PICTBridge:IDPrint {DEFLT|OFF|ON}
```

**Related Commands**

**Arguments**

- **DEFLT** is the default setting of the printer.
- **ON** enables the oscilloscope model and serial number print on the print output.
- **OFF** disables the oscilloscope model and serial number print on the print output.
Examples

PICTBridge:IDPRINT ON enables the printing of oscilloscope model and serial number on the print output.

PICTBridge:IDPRINT? might return PICTBridge:IDPRINT OFF indicating that the ID print is disabled on the print output.

PICTBridge:IMAGESize

Sets or returns the image print size.

Group

PictBridge

Syntax

PICTBridge:IMAGESize
{DEFLT|IN2P5BY3P25|L|IN4BY6|L2|IN8BY10|L4|E|CARD|HAGAKIPcard|CM6BY8|CM7BY10|CM9BY13|CM10BY15|CM13BY18|CM15BY21|CM18BY24|A4|LETTER}

Arguments

DEFLT for a default image print size.

IN2P5BY3P25 for a 2_5*3_25 image print size.

L for a 3_5*5 image print size.

IN4BY6 for a 4*6 image print size.

L2 for a 5*7 image print size.

IN8BY10 for a 8*10 image print size.

L4 for a 254 MM*178 MM image print size.

E for a 110 MM*74 MM image print size.

CARD for a 89 MM*55 MM image print size.

HAGAKIPcard for a 100 MM*148 MM image print size.

CM6BY8 for a 6 CM*8 CM image print size.

CM7BY10 for a 7 CM*10 CM image print size.

CM9BY13 for a 9 CM*13 CM image print size.

CM10BY15 for a 10 CM*15 CM image print size.

CM13BY18 or a 13 CM*18 CM image print size.

CM15BY21 for a 15 CM*21 CM image print size.

CM18BY24 for a 18 CM*24 CM image print size.

A4 for a A4 image print size.

LETTER for a Letter image print size.
Examples
PICTBRIDGE:IMAGESIZE CARD sets the image print size to 89 MM* 55 MM.
PICTBRIDGE:IMAGESIZE? might return PICTBRIDGE:IMAGESIZE DEFLT indicating the default image print size.

PICTBridge:PAPERSize

Sets the output print paper size.

Group
PictBridge

Syntax
PICTBridge:PAPERSize
DEFLT|L|L2|HAGAKIPCARD|MM54BY86|MM100BY150|IN4BY6|IN8BY10|LETTER|IN11BY17|A0|A1|A2|A3|A4|A5|A6|A7|A8|B4|B5|B6|B7|B8|B9|ROLL89MM|ROLL100MM|ROLL210MM

Arguments
DEFLT for a default paper size.
L for a paper size L
L2 for a paper size 2L.
HAGAKIPCARD for a paper size Hagaki.
MM54BY86 for a card paper size.
MM100BY150 for paper size of 100*150 MM.
IN4BY6 for a paper size of 4*6.
IN8BY10 for a paper size of 8*10.
LETTER for a letter paper size.
IN11BY17 for a paper size of 11*17.
A0 for an A0 paper size.
A1 for an A1 paper size.
A2 for an A2 paper size.
A3 for an A3 paper size.
A4 for an A4 paper size.
A5 for an A5 paper size.
A6 for an A6 paper size.
A7 for an A7 paper size.
A8 for an A8 paper size.
A9 for a A9 paper size.
B0 for a B0 paper size.
B1 for a B1 paper size.
B2 for a B2 paper size.
B3 for a B3 paper size.
B4 for a B4 paper size.
B5 for a B5 paper size.
B6 for a B6 paper size.
B7 for a B7 paper size.
B8 for a B8 paper size.
B9 for a B9 paper size.
ROLL89MM for a 89 MM Roll paper size.
ROLL127MM for a 127 MM Roll paper size.
ROLL100MM for a 100 MM Roll paper size.
ROLL210MM for a 210 MM Roll paper size.

**Examples**

PICTBRIDGE:PAPERSIZE L sets the paper size to L.

PICTBRIDGE:PAPERSIZE? might return PICTBRIDGE:PAPERSIZE DEFLT indicating the paper size is set to Default.

**PICTBridge:PAPERType**

Sets or returns the paper type.

**Group**
PictBridge

**Syntax**

PICTBridge:PAPERType {DEFLT|PLAIN|PHOTO|FASTPHOTO}

**Arguments**

DEFLT for a default print paper type.

PLAIN for a plain print paper type.

PHOTO for a photo print paper type.

FASTPHOTO for a fastphoto print paper type.
**Examples**  
PICTBRIDGE:PAPERTYPE PLAIN sets the paper type to Plain.

PICTBRIDGE:PAPERTYPE? might return PICTBRIDGE:PAPERTYPE DEFLT indicating the default paper type.

**PICTBridge:PRINTQual**

Sets or returns the output print quality.

**Group**  
PictBridge

**Syntax**  
PICTBridge:PRINTQual \{DEFLT|NRMAL|FINE|DRAFT\}

**Arguments**  
DEFLT for the default quality print.

NRMAL for a normal quality print.

FINE for a fine quality print.

DRAFT for a draft quality print.

**Examples**  
PICTBRIDGE:PRINTQUAL FINE sets the print quality to Fine.

PICTBRIDGE:PRINTQUAL? might return PICTBRIDGE:PRINTQUAL DEFLT indicating the default quality print.

**POWer:CURRENTSOurce**

Sets or returns the current source for the power application.

**Conditions**  
This command requires a DPO4PWR application module.

**Group**  
Power

**Syntax**  
POWer:CURRENTSOurce \{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4\}

POWer:CURRENTSOurce?

**Arguments**  
CH1–CH4 sets an analog channel as the current source.

REF1–REF4 sets a reference as the current source.
**Examples**

POWER:CURRENTSOURCE CH1 sets the current source as CH1.

POWER:CURRENTSOURCE? might return POWER:CURRENTSOURCE CH2 indicating that CH2 is the current source.

**POWer:DISplay**

Sets or returns the display state for the power application. This is the equivalent to pressing the Test button and then selecting the power application. The same control is provided for each application.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

POWer:DISplay {OFF|ON|0|1}

POWer:DISplay?

**Arguments**

OFF or 0 turns off the display settings.

ON or 1 turns on the display settings.

**Examples**

POWER:DISPLAY 1 turns on the display.

POWER:DISPLAY? might return :POWER:DISPLAY 0 indicating that the display is off.

**POWer:GATESOurce**

Sets or returns the gate source for the power application.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

POWer:GATESOurce {CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|NONE}

POWer:GATESOurce?

**Arguments**

CH1–CH4 sets an analog channel as the gate source.

REF1–REF4 sets a reference as the gate source.
NONE is set when the gate source is not used in the application.

**Examples**

POWER:GATESOURCE CH1 sets the gated source as CH1.

POWER:GATESOURCE? might return POWER:GATESOURCE CH2 indicating that CH2 is the gated source.

**POWer:GATing**

Sets or returns the power application gating.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

POWer:GATing {OFF|SCREen|CURSor}

POWer:GATing?

**Arguments**

OFF turns off measurement gating (full 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.

**Examples**

POWER:GATING CURSor sets cursor gating on the oscilloscope.

POWER:GATING? might return POWER:GATING OFF indicating that gating is turned off on the oscilloscope.

**POWer:HARMonics:DISPlay:SELect**

Sets or returns the harmonics to be displayed when the harmonics standard is NONe.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

POWer:HARMonics:DISPlay:SELect {ODD|EVEN|ALL}

POWer:HARMonics:DISPlay:SELect?
**Arguments**

ODD to display only odd harmonics.

EVEN to display only even harmonics.

ALL to display both odd and even harmonics.

**Examples**

POWER:HARMONICS:DISPLAY:SELECT EVEN displays only even harmonics.

POWER:HARMONICS:DISPLAY:SELECT? might return :POWER:HARMONICS:DISPLAY:SELECT ALL indicating that both odd and even harmonics are displayed.

**POWER:HARMONICS:DISPLAY:TYPE**

Sets or returns the display type for harmonics tests.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

POWER:HARMONICS:DISPLAY:TYPE {GRAph|TABle}

POWER:HARMONICS:DISPLAY:TYPE?

**Arguments**

GRAph displays harmonic tests results in graphical format.

TABle displays harmonic tests results in tabular format.

**Examples**

POWER:HARMONICS:DISPLAY:TYPE GRAph sets the display type to graphical.


**POWER:HARMONICS:FREQRef**

Sets or returns the frequency reference used when the harmonic standard is None.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power
Syntax

POWer:HARMonics:FREQRef
{VOLTage|CURRent|HARMSOURce|FIXEDFREQuency}
POWer:HARMonics:FREQRef?

Arguments

VOLTage to use a voltage waveform as the frequency reference.
CURRent to use a current waveform as the frequency reference.
HARMSOURce to use a harmonic source waveform as the frequency reference.
FIXEDFREQuency to use a fixed frequency value instead of a waveform for the frequency reference.

Examples

POWER:HARMONICS:FREQRef VOLTage sets voltage source as the frequency reference.

POWER:HARMONICS:FREQRef? might return :POWER:HARMONICS:FREQRef FIXEDFREQ indicating that the frequency reference is a fixed value.

POWer:HARMonics:FREQRef:FIXEDFREQValue

Sets or returns the frequency value when the :FREQRef selection is FIXEDFREQuency.

Conditions

This command requires a DPO4PWR application module.

Group

Power

Syntax

POWer:HARMonics:FREQRef:FIXEDFREQValue <NR3>
POWer:HARMonics:FREQRef:FIXEDFREQValue?

Arguments

<NR3> is the fixed frequency value.

Examples

POWER:HARMONICS:FREQRef:FIXEDFREQValue 100 sets the frequency to 100 Hz.

POWER:HARMONICS:FREQRef:FIXEDFREQValue? might return :POWER:HARMONICS:FREQRef:FIXEDFREQValue 60 indicating that the fixed frequency value is set to 60 Hz.
POWer:HARMonics:IEC:CLAss

Sets or returns the equipment class for IEC harmonics.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

```
POWer:HARMonics:IEC:CLAss {A|B|C1|C2|C3|D}
POWer:HARMonics:IEC:CLAss?
```

**Arguments**

- **A** specifies Class A Equipment.
- **B** specifies Class B Equipment.
- **C1** specifies Class C Equipment that use Table 1 limits of the IEC standard.
- **C2** specifies Class C Equipment that use Table 2 limits of the IEC standard.
- **C3** specifies Class C Equipment that use Table 3 limits of the IEC standard.
- **D** specifies Class D Equipment.

**Examples**

```plaintext

POWER:HARMONICS:IEC:CLASS? might return

:POWER:HARMONICS:IEC:CLASS A indicating that the Equipment Class is A
for IEC harmonics.
```

POWer:HARMonics:IEC:FILter

Sets or returns the enabled state for filtering of IEC harmonics.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

```
POWer:HARMonics:IEC:FILter {OFF|ON|0|1}
POWer:HARMonics:IEC:FILter?
```

**Arguments**

- **ON** or 1 enables filtering of IEC harmonics.
- **OFF** or 0 disables filtering of IEC harmonics.
### Examples


### POWER:HARMONICS:IEC:FUNDamental

Sets or returns the rated fundamental current for IEC harmonics. Valid values ranges from 0 to 16 in increments of 0.1. The unit is ampere.

#### Conditions

This command requires a DPO4PWR application module.

#### Group

Power

#### Syntax

POWER:HARMONICS:IEC:FUNDamental <NR3>

POWER:HARMONICS:IEC:FUNDamental?

#### Arguments

<NR3> is the fundamental current in amperes.

#### Examples

POWER:HARMONICS:IEC:FUNDAMENTAL 1 sets the fundamental current to 1.0000 A.


### POWER:HARMONICS:IEC:GROUPing

Sets or returns the enabled state for grouping of IEC harmonics.

#### Conditions

This command requires a DPO4PWR application module.

#### Group

Power

#### Syntax

POWER:HARMONICS:IEC:GROUPing {OFF|ON|1|0}

POWER:HARMONICS:IEC:GROUPing?
Arguments  
ON or 1 enables grouping of IEC harmonics.
OFF or 0 disables grouping of IEC harmonics.

Examples  

POWer:HARMonics:IEC:INPUTPOWer  
Sets or returns the class D rated input power for IEC harmonics.

Conditions  
This command requires a DPO4PWR application module.

Group  
Power

Syntax  
POWer:HARMonics:IEC:INPUTPOWer <NR3>
POWer:HARMonics:IEC:INPUTPOWer?

Arguments  
<NR3> is the class D input power. Valid values ranges from 0 to 600 in increments of 10. The unit of measure is watt.

Examples  
POWER:HARMONICS:IEC:INPUTPOWER 600 sets the class D input power to 600 W.
POWER:HARMONICS:IEC:INPUTPOWER? might return :POWER:HARMONICS:IEC:INPUTPOWER 100 indicating that the class D input power is set to 100 W.

POWer:HARMonics:IEC:LINEFREQuency  
Sets or returns the line frequency for the IEC standard.

Conditions  
This command requires a DPO4PWR application module.

Group  
Power
### Commands Listed in Alphabetical Order

#### Syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
</tr>
</thead>
</table>
POWer:HARMonics:IEC:LINEFREQuency? |

#### Arguments

<NR3> is the line frequency. The valid values are 50 and 60.

#### Examples

POWER:HARMONICS:IEC:LINEFREQUENCY 50 sets the line frequency to 50 Hz.

POWER:HARMONICS:IEC:LINEFREQUENCY? might return

:POWER:HARMONICS:IEC:LINEFREQUENCY 60 indicating that the
line frequency value is set to 60 Hz.

#### POWer:HARMonics:IEC:OBSPERiod

Sets or returns the IEC observation period.

#### Conditions

This command requires a DPO4PWR application module.

#### Group

Power

#### Syntax

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
</tr>
</thead>
</table>
POWer:HARMonics:IEC:OBSPERiod? |

#### Arguments

<NR3> is the IEC observation period. Valid values ranges from 0.2 to 10 s.

#### Examples

POWER:HARMONICS:IEC:OBSPERIOD 3 sets the IEC observation period to 3.0000 s.

POWER:HARMONICS:IEC:OBSPERIOD? might return

:POWER:HARMONICS:IEC:OBSPERIOD 10 indicating that the IEC observation
period is set to 10 s.

#### POWer:HARMonics:IEC:POWERFACTOR

Sets or returns the rated power factor for IEC harmonics.

#### Conditions

This command requires a DPO4PWR application module.

#### Group

Power
Commands Listed in Alphabetical Order

**POWer:HARMonics:IEC:POWERFACtor**

**Syntax**
- `POWer:HARMonics:IEC:POWERFACtor <NR3>`
- `POWer:HARMonics:IEC:POWERFACtor?`

**Arguments**
- `<NR3>` is the power factor. Valid values range from 0 to 1 in increments of 0.1.

**Examples**
- `POWER:HARMONICS:IEC:POWERFACTOR 1` sets the power factor to 1.0000.
- `POWER:HARMONICS:IEC:POWERFACTOR?` might return `:POWER:HARMONICS:IEC:POWERFACTOR 900.0000E-03` indicating that the power factor is set to 0.9.

**POWer:HARMonics:MIL:FUNDamental:CALCmethod**

Sets or returns the measurement method for the MIL harmonics fundamental current for use in calculating limits.

**Conditions**
- This command requires a DPO4PWR application module.

**Group**
- Power

**Syntax**
- `POWer:HARMonics:MIL:FUNDamental:CALCmethod {MEAS|USER}`
- `POWer:HARMonics:MIL:FUNDamental:CALCmethod?`

**Arguments**
- `MEAS` specifies that the value of the fundamental current used in calculating limits is measured.
- `USER` specifies that the value of the fundamental current used in calculated limits is user defined.

**Examples**
- `POWER:HARMONICS:MIL:FUNDAMENTAL:CALCMETHOD USER` sets the measurement method to User.

**POWer:HARMonics:MIL:FUNDamental:USER:CURRent**

Sets or returns RMS amperes for USER CALCmethod.

**Conditions**
- This command requires a DPO4PWR application module.
**Group**  
Power

**Syntax**  
`POWer:HARMonics:MIL:FUNDamental:USER:CURRent?`

**Arguments**  
`<NR3>` is the current in amperes for USER CALCmethod.

**Examples**  

---

**POWer:HARMonics:MIL:LINEFREQuency**  
Sets or returns the line frequency for MIL-STD-1399 harmonics tests. Valid values are 60 or 400 Hz.

**Conditions**  
This command requires a DPO4PWR application module.

**Group**  
Power

**Syntax**  
`POWer:HARMonics:MIL:LINEFREQuency <NR3>`  
`POWer:HARMonics:MIL:LINEFREQuency?`

**Arguments**  
`<NR3>` is the line frequency for MIL standard.

**Examples**  
`POWER:HARMONICS:MIL:LINEFREQUENCY 60` sets the MIL line frequency to 60 Hz.  
`POWER:HARMONICS:MIL:LINEFREQUENCY?` might return `:POWER:HARMONICS:MIL:LINEFREQUENCY 400` indicating that the MIL line frequency is set to 400 Hz.

---

**POWer:HARMonics:MIL:POWERLEVel**  
Sets or returns the power level for calculating limits for MIL-STD-1399 harmonics tests.
Conditions
This command requires a DPO4PWR application module.

Group
Power

Syntax
POWeR:HARMonics:MIL:POWERLeVel \{LOW|HIGH\}
POWeR:HARMonics:MIL:POWERLeVel?

Arguments
LOW specifies low power level for MIL-STD-1399 harmonics tests.
HIGH specifies high power level for MIL-STD-1399 harmonics tests.

Examples
POWER:HARMONICS:MIL:POWERLEVEL HIGH sets the MIL power level to High.
POWER:HARMONICS:MIL:POWERLEVEL? might return
:POWER:HARMONICS:MIL:POWERLEVEL LOW indicating that the MIL power level is set to Low.

POWeR:HARMonics:NR_HARMonics
Sets or returns the number of harmonics (value ranging from 20 to 400) when the harmonics standard is NONe.

Conditions
This command requires a DPO4PWR application module.

Group
Power

Syntax
POWeR:HARMonics:NR_HARMonics <NR3>
POWeR:HARMonics:NR_HARMonics?

Arguments<br> <NR3> is the number of harmonics.

Examples
POWER:HARMONICS:NR_HARMonics 100 sets the number of harmonics to 100.
POWER:HARMONICS:NR_HARMonics? might return
:POWER:HARMONICS:NR_HARMonics 40 indicating that the number of harmonics is set to 40.

POWeR:HARMonics:RES ults:HAR<1-400>:FREQuency? (Query Only)
Returns the frequency of the harmonic.
NOTE. The command returns NA, if the current harmonic standard is set to a new standard other than the returned query.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:HARMonics:RESults:HAR<1-400>:FREQuency?

**Examples**
POWer:HARMonics:RESULTS:HAR400:FREQUENCY? might return:
:POWer:HARMonics:RESULTS:HAR400:FREQUENCY 24000 indicating that the harmonic frequency is set to 24000.

**POWer:HARMonics:RESults:HAR<1-400>:IECMAX? (Query Only)**

The IEC Standard specifies harmonics measurements to be computed in time windows, with each time window being nominally 200 ms. This returns the maximum of the RMS magnitude of the harmonic, computed across successive 200 ms time windows within an observation period entered by the user.

NOTE. The command returns NA, if the current harmonic standard is set to a new standard other than the returned query.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:HARMonics:RESults:HAR<1-400>:IECMAX?

**Examples**
POWer:HARMonics:RESULTS:HAR400:IECMAX? might return:
:POWer:HARMonics:RESULTS:HAR400:IECMAX 1.0 indicating that the maximum value is 1.0 A for the 400th harmonic.

**POWer:HARMonics:RESults:HAR<1-400>:LIMit? (Query Only)**

The IEC and MIL standards specify a limit for each harmonic magnitude. Returns the limit in absolute units, or as a percentage of the fundamental as specified by
the standard. IEC Class C (Table 2) and MIL standards specify the limit as a percentage of the fundamental.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
```
POWer:HARMonics:RESults:HAR<1-400>:LIMit?
```

**Examples**
```
POWer:HARMonics:RESults:HAR3:LIMIT?
```
might return
```
:POWer:HARMonics:RESults:HAR3:LIMIT 2.30
```
indicating that the harmonic magnitude limit for IEC standard for the 3rd harmonic is set to 2.30 A.

**POWer:HARMonics:RESults:HAR<1-400>:PHASE? (Query Only)**

Returns the phase of the harmonic in degrees. The phase is measured relative to the zero-crossing of the reference waveform. When there is no reference waveform, the phase is relative to the fundamental component.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
```
POWer:HARMonics:RESults:HAR<1-400>:PHASE?
```

**Examples**
```
POWer:HARMonics:RESults:HAR400:PHASE?
```
might return
```
:POWer:HARMonics:RESults:HAR400:PHASE 0
```
indicating that the harmonic phase is set to 0 in degrees.

**POWer:HARMonics:RESults:HAR<1-400>:RMS:ABSolute? (Query Only)**

Returns the RMS magnitude of the harmonic in absolute units.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power
Syntax  
POWer:HARMonics:RESults:HAR<1-400>:RMS:ABSolute?

Examples  
:POWER:HARMonics:RESults:HAR400:RMS:ABSolute 1.0 indicating that 
the RMS magnitude of the harmonic is set to 1.0 A.

POWer:HARMonics:RESults:HAR<1-400>:RMS:PERCent? (Query Only)

Returns the RMS magnitude of the harmonic expressed as a percentage of the 
fundamental.

Conditions  
This command requires a DPO4PWR application module.

Group  
Power

Syntax  
POWer:HARMonics:RESults:HAR<1-400>:RMS:PERCent?

Examples  
POWer:HARMonics:RESults:HAR400:RMS:PERCent? might return 
:POWER:HARMonics:RESults:HAR400:RMS:PERCent 33 indicating that the 
RMS magnitude of the harmonic is set to 33%.

POWer:HARMonics:RESults:HAR<1-400>:TEST:IEC:CLASSALIMit? (Query Only)

Specifies if the IEC Class A higher harmonic limit (and conditions) are met.

**NOTE.** The command returns NA if the standard does not specify a limit or the 
equipment is not class A.

Conditions  
This command requires a DPO4PWR application module.

Group  
Power

Syntax  
POWer:HARMonics:RESults:HAR<1-400>:TEST:IEC:CLASSALIMit?

Returns  
PASS, FAIL, or NA.

Specifies if the Normal IEC harmonic limits are met.

**NOTE.** The command returns NA if the standard does not specify a limit.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:HARMonics:RESults:HAR<1-400>:TEST:IEC:NORMAL?

**Returns**
PASS, FAIL, or NA.

POWer:HARMonics:RESults:HAR<1-400>:TEST:IEC:POHCLIMit? (Query Only)

Specifies if the higher harmonic limit (and conditions) for the 21st and higher order odd harmonics are met.

**NOTE.** The command returns NA if the limit does not apply for a specific harmonic.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:HARMonics:RESults:HAR<1-400>:TEST:IEC:POHCLIMit?

**Returns**
PASS, FAIL, or NA.

POWer:HARMonics:RESults:HAR<1-400>:TEST:MIL:NORMAL? (Query Only)

Returns the test result for the specified harmonic for the MIL-STD-1399 testing standard.

This query is analogous to that for the IEC 61000-3-2 standard POWer:HARMonics:RESults:HAR<1-400>:TEST:IEC:NORMAL? command.
NOTE. The command returns NA if the standard does not specify a limit for the specific harmonic.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
`POWer:HARMonics:RESults:HAR<1-400>:TEST:MIL:NORMAL?`

**Returns**
PASS, FAIL, or NA.

**POWer:HARMonics:RESults:IEC:FUNDamental? (Query Only)**
Returns the IEC fundamental current used in calculating limits.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
`POWer:HARMonics:RESults:IEC:FUNDamental?`

**Examples**
`POWER:HARMONICS:RESULTS:IEC:FUNDAMENTAL 1.0` indicating that the IEC fundamental frequency is set to 1.0 A.

**POWer:HARMonics:RESults:IEC:HARM3ALTernate? (Query Only)**
Returns the IEC harmonics test result for the 3rd harmonic.

NOTE. The command returns NA if the limit does not apply.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
`POWer:HARMonics:RESults:IEC:HARM3ALTernate?`
Commands Listed in Alphabetical Order

**Returns**
PASS, FAIL, or NA.

**Examples**
POWER:HARMONICS:RESULTS:IEC:HARM3ALTERNATE PASS indicating the
test result.

**POWER:HARMONICS:RESULTS:IEC:HARM5ALTERNATE?** (Query Only)
Returns the overall harmonics test result for the 5th harmonic.

*NOTE. The command returns NA the harmonic limit does not apply.*

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWER:HARMONICS:RESULTS:IEC:HARM5ALTERNATE?

**Returns**
PASS, FAIL, or NA.

**Examples**
POWER:HARMONICS:RESULTS:IEC:HARM5ALTERNATE? might return
POWER:HARMONICS:RESULTS:IEC:HARM5ALTERNATE PASS indicating the
test result.

**POWER:HARMONICS:RESULTS:IEC:POHC?** (Query Only)
Returns the IEC POHC measurement.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWER:HARMONICS:RESULTS:IEC:POHC?

**Examples**
POWER:HARMONICS:RESULTS:IEC:POHC 0.5 indicating that the
IEC POHC is set to 0.5 A.
POWer:HARMonics:RESults:IEC:POHL? (Query Only)

Returns the IEC POHL measurement.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:HARMonics:RESults:IEC:POHL?

**Examples**
POWer:HARMonics:RESults:IEC:POHL? might return
:POWER:HARMONICS:RESULTS:IEC:POHL 0.5 indicating that the IEC POHL is set to 0.5 A.

POWer:HARMonics:RESults:IEC:POWer? (Query Only)

Returns the IEC input power that is used to calculate limits.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:HARMonics:RESults:IEC:POWer?

**Examples**
POWer:HARMonics:RESults:IEC:POWer? might return
:POWER:HARMONICS:RESULTS:IEC:POWER 5 indicating that the IEC power is set to 5 W.

POWer:HARMonics:RESults:IEC:POWERFactor? (Query Only)

Returns the IEC power factor measurement.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:HARMonics:RESults:IEC:POWERFactor?
Examples

that the IEC power factor is set to 9.1900E+37.

POWer:HARMonics:RESults:PASSFail? (Query Only)

Returns the overall harmonics test result: PASS, FAIL or NA.

Conditions

This command requires a DPO4PWR application module.

Group

Power

Syntax

POWER:HARMONICS:RESULTS:PASSFail?

Examples

POWER:HARMONICS:RESULTS:PASSFail? might return
:POWER:HARMONICS:RESULTS:PASSFail PASS indicating that the
harmonics test passed.

POWer:HARMonics:RESults:RMS? (Query Only)

Returns the root mean square value of the harmonics source waveform.

Conditions

This command requires a DPO4PWR application module.

Group

Power

Syntax

POWER:HARMONICS:RESULTS:RMS?

Examples

POWER:HARMONICS:RESULTS:RMS? might return
:POWER:HARMONICS:RESULTS:RMS 1.0 indicating that the harmonics source
waveform RMS is set to 1.0.

POWer:HARMonics:RESults:SAVe (No Query Form)

Saves the harmonic results to the specified file in CSV format.

Conditions

This command requires a DPO4PWR application module.
Group: Power

Syntax: POWER:HARMOnics:RESults:SAVe <String>

**POWer:HARMOnics:RESults:THDF? (Query Only)**

Returns the Total Harmonic Distortion (THD) in percentage, measured as a ratio to the RMS value of the fundamental component of the source waveform.

Conditions: This command requires a DPO4PWR application module.

Syntax: POWER:HARMOnics:RESults:THDF?

Examples:

```
POWER:HARMONICS:RESULTS:THDF? might return
:POWER:HARMONICS:RESULTS:THDF 40 indicating that the THDF is set to 40.
```

**POWer:HARMOnics:RESults:THDR? (Query Only)**

Returns the Total Harmonic Distortion (THD) in percentage, measured as a ratio to the RMS value of the source waveform.

Conditions: This command requires a DPO4PWR application module.

Syntax: POWER:HARMOnics:RESults:THDR?

Examples:

```
POWER:HARMONICS:RESULTS:THDR? might return
:POWER:HARMONICS:RESULTS:THDR 40 indicating that the THDR is set to 40.
```

**POWer:HARMOnics:SOURce**

Sets or returns the source waveform for harmonics tests. The voltage source waveform is specified using the POWER:VOLTAGE:SOURce command and the current waveform is specified using the POWER:CURRENT:SOURce command.
**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
```
POWER:HARMONICS:SOURce {VOLTage|CURRent}
POWER:HARMONICS:SOURce?
```

**Arguments**
- **VOLTage** specifies voltage source waveform for harmonic tests.
- **CURRent** specifies current source waveform for harmonic tests.

**Examples**
- `POWER:HARMONICS:SOURce VOLTage` sets voltage source for harmonic tests.
- `POWER:HARMONICS:SOURce?` might return `:POWER:HARMONICS:SOURce CURRENT` indicating that the harmonic source is set to current.

**POWER:HARMONICS:STANDARD**
Sets or returns the standard for harmonics tests.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
```
POWER:HARMONICS:STANDard {NONE|IEC|MIL}
POWER:HARMONICS:STANDard?
```

**Arguments**
- **NONE** sets no standard for harmonic tests.
- **IEC** sets IEC 61000-3-2 standard for harmonic tests.
- **MIL** sets MIL1399 standard for harmonic tests.

**Examples**
- `POWER:HARMONICS:STANDard IEC` sets IEC standard for harmonic tests.

**POWER:INDICATORS**
Sets or returns the state of the measurement indicators for the power application.
### POWER:INDICATORS

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**

```
POWER:INDICATORS {OFF|ON|0|1}
POWER:INDICATORS?
```

**Arguments**
OFF or 0 turns off the measurement indicators.
ON or 1 turns on the measurement indicators.

**Examples**

```
POWER:INDICATORS 1 turns on the indicator.
POWER:INDICATORS? might return :POWER:INDICATORS 0 indicating that the indicator is off.
```

### POWER:MODULATION:SOURce

Sets or returns the source waveform for modulation tests. The voltage source waveform is specified using the `POWER:VOLTAGE:SOURce` command and the current waveform is specified using the `POWER:CURRENT:SOURce` command.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**

```
POWER:MODULATION:SOURce {VOLTage|CURRent}
POWER:MODULATION:SOURce?
```

**Arguments**
VOLTage specifies voltage source waveform for modulation tests.
CURRent specifies current source waveform for modulation tests.

**Examples**

```
POWER:MODULATION:SOURce CURRent sets the modulation source to Current.
POWER:MODULATION:SOURce? might return :POWER:MODULATION:SOURce VOLT indicating that the modulation source is set to Voltage.
```
**POWer:MODulation:TYPe**

Sets or returns the modulation type.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
```
POWer:MODulation:TYPe
{PWidth|NWidth|PERIod|PDUty|NDUty|FREQuency}
POWer:MODulation:TYPe?
```

**Arguments**
- **PWidth** (positive width) is the distance (time) between the middle reference (default = 50%) amplitude points of a positive pulse. The measurement is made on all the cycles in the waveform or gated region.

- **NWidth** (negative width) measurement is the distance (time) between the middle reference (default = 50%) amplitude points of a negative pulse. The measurement is made on all the cycles in the waveform or gated region.

- **PERIod** is the time required to complete the first cycle in a waveform or the gated region. The time is measured between the mid reference (default being 50%) amplitude points of the waveform.

- **PDUty** (positive duty cycle) is the ratio of the positive pulse width to the signal period, expressed as a percentage. It is measured on all the cycles in the waveform or gated region.

  \[ \text{Positive Duty Cycle} = \left( \frac{\text{Positive Width}}{\text{Period}} \right) \times 100\% \]

- **NDUty** (negative duty cycle) is the ratio of the negative pulse width to the signal period, expressed as a percentage. The duty cycle is measured on all the cycles in the waveform or gated region.

  \[ \text{Negative Duty Cycle} = \left( \frac{\text{Negative Width}}{\text{Period}} \right) \times 100\% \]

- **FREQuency** measures all the cycles in the waveform or gated region. Frequency is the reciprocal of the period and is measured in hertz (Hz), where 1 Hz = 1 cycle per second.

  \[ \text{Frequency} = \frac{1}{\text{Period}} \]

**Examples**
- `POWER:MODULATION:TYPE NWIDTH` sets the modulation type to Negative Width.

- `POWER:MODULATION:TYPE?` might return `POWER:MODULATION:TYPE PWidth` indicating that the modulation type is set to Positive Width.
**POWer:QUALity:APPpwr? (Query Only)**

_Returns the apparent power measurement._

**Conditions**  
This command requires a DPO4PWR application module.

**Group**  
Power

**Syntax**  
`POWer:QUALity:APPpwr?`

**Examples**  
`POWER:QUALITY:APPPWR?` might return `:POWER:QUALITY:APPPwr 100` indicating that the apparent power value is set to 100 VA.

**POWer:QUALity:DISplay:APPpwr**

_Sets or returns the display state for the apparent power readout._

**Conditions**  
This command requires a DPO4PWR application module.

**Group**  
Power

**Syntax**  
`POWer:QUALity:DISplay:APPpwr {OFF|ON|0|1}`  
`POWer:QUALity:DISplay:APPpwr?`

**Arguments**  
OFF or 0 turns off the apparent power display.  
ON or 1 turns on the apparent power display.

**Examples**  
`POWER:QUALITY:DISPLAY:APPPWR 1` turns on the apparent power display.  
`POWER:QUALITY:DISPLAY:APPPWR?` might return `:POWER:QUALITY:DISPLAY:APPPwr 0` indicating that the apparent power display is off.

**POWer:QUALity:DISplay:FREQuency**

_Sets or returns the display state for the frequency readout._

**Conditions**  
This command requires a DPO4PWR application module.
Commands Listed in Alphabetical Order

<table>
<thead>
<tr>
<th>Group</th>
<th>Power</th>
</tr>
</thead>
</table>
| Syntax    | `POWer:QUALity:DISplay:FREQuency {OFF|ON|0|1}`  
            | `POWer:QUALity:DISplay:FREQuency?` |
| Arguments | OFFSET or 0 turns off the frequency display.  
            | ON  or 1 turns on the frequency display. |
| Examples  | `POWER:QUALITY:DISPLAY:FREQUENCY 1` turns on the frequency readout display.  
            | `POWER:QUALITY:DISPLAY:FREQUENCY?` might return  
            | `:POWER:QUALITY:DISPLAY:FREQUENCY 0` indicating that the  
            | frequency readout display is off. |

**POWer:QUALity:DISplay:ICRESTfactor**

Sets or returns the display state for the current crest factor readout.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>This command requires a DPO4PWR application module.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Power</td>
</tr>
</tbody>
</table>
| Syntax    | `POWer:QUALity:DISplay:ICRESTfactor {OFF|ON|0|1}`  
            | `POWer:QUALity:DISplay:ICRESTfactor?` |
| Arguments | OFF or 0 turns off the current crest factor display.  
            | ON or 1 turns on the current crest factor display. |
| Examples  | `POWER:QUALITY:DISPLAY:ICRESTFACTOR 1` turns on the current crest factor display.  
            | `POWER:QUALITY:DISPLAY:ICRESTFACTOR?` might return  
            | `:POWER:QUALITY:DISPLAY:ICRESTFACTOR 0` indicating that the  
            | current crest factor display is off. |

**POWer:QUALity:DISplay:IRMS**

Sets or returns the display state for the rms current readout.
**Conditions**  
This command requires a DPO4PWR application module.

**Group**  
Power

**Syntax**  
\[ \text{POWer:QUALity:DISplay:IRMS} \{\text{OFF|ON|0|1}\} \]
\[ \text{POWer:QUALity:DISplay:IRMS?} \]

**Arguments**  
OFF or 0 turns off the rms current display.
ON or 1 turns on the rms current display.

**Examples**  
\[ \text{POWer:QUALity:DISplay:IRMS} 1 \] turns on the rms current display.
\[ \text{POWer:QUALity:DISplay:IRMS?} \] might return 
\[ :\text{POWer:QUALity:DISplay:IRMS} 0 \] indicating that the rms current display is off.

**POWer:QUALity:DISplay:PHASEangle**  
Sets or returns the display state for the phase angle readout.

**Conditions**  
This command requires a DPO4PWR application module.

**Group**  
Power

**Syntax**  
\[ \text{POWer:QUALity:DISplay:PHASEangle} \{\text{OFF|ON|0|1}\} \]
\[ \text{POWer:QUALity:DISplay:PHASEangle?} \]

**Arguments**  
OFF or 0 turns off the phase angle display.
ON or 1 turns on the phase angle display.

**Examples**  
\[ \text{POWer:QUALity:DISplay:PHASEangle} 1 \] turns on the phase angle display.
\[ \text{POWer:QUALity:DISplay:PHASEangle?} \] might return 
\[ :\text{POWer:QUALity:DISplay:PHASEangle} 0 \] indicating that the phase angle display is off.

**POWer:QUALity:DISplay:POWERFACtor**  
Sets or returns the display state for the power factor readout.
**POWer:QUALity:DISplay:POWERFACtor**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

```
POWer:QUALity:DISplay:POWERFACtor {OFF|ON|0|1}
POWer:QUALity:DISplay:POWERFACtor?
```

**Arguments**

OFF or 0 turns off the power factor display.

ON or 1 turns on the power factor display.

**Examples**

```
POWER:QUALITY:DISPLAY:POWERFACTOR 1
```

turns on the power factor display.

```
POWER:QUALITY:DISPLAY:POWERFACTOR?
```

might return

```
:POWER:QUALITY:DISPLAY:POWERFACTOR 0
```

indicating that the power factor display is off.

**POWer:QUALity:DISplay:REACTpwr**

Sets or returns the display state for the reactor power readout.

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

```
POWer:QUALity:DISplay:REACTpwr {OFF|ON|0|1}
POWer:QUALity:DISplay:REACTpwr?
```

**Arguments**

OFF or 0 turns off the reactor power display.

ON or 1 turns on the reactor power display.

**Examples**

```
POWER:QUALITY:DISPLAY:REACTPWR 1
```

turns on the reactor power display.

```
POWER:QUALITY:DISPLAY:REACTPWR?
```

might return

```
:POWER:QUALITY:DISPLAY:REACTPWR 0
```

indicating that the reactor power display is off.
**POWer:QUALity:DISplay:TRUEpwr**

Sets or returns the display state for the true power readout.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

POWer:QUALity:DISPLAY:TRUEpwr {OFF|ON|0|1}

POWer:QUALity:DISPLAY:TRUEpwr?

**Arguments**

OFF or 0 turns off the true power display.

ON or 1 turns on the true power display.

**Examples**

POWER:QUALITY:DISPLAY:TRUEPWR 1 turns on the true power display.

POWER:QUALITY:DISPLAY:TRUEPWR? might return

:POWER:QUALITY:DISPLAY:TRUEPWR 0 indicating that the true power display is off.

**POWer:QUALity:DISplay:VCRESTfactor**

Sets or returns the display state for the voltage crest factor readout.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

POWer:QUALity:DISPLAY:VCRESTfactor {OFF|ON|0|1}

POWer:QUALity:DISPLAY:VCRESTfactor?

**Arguments**

OFF or 0 turns off the voltage crest factor display.

ON or 1 turns on the voltage crest factor display.

**Examples**

POWER:QUALITY:DISPLAY:VCRESTFACTOR 1 turns on the voltage crest factor display.
POWER:QUALITY:DISPLAY:VCRESTFACTOR? might return :POWER:QUALITY:DISPLAY:VCRESTFACTOR 0 indicating that the voltage crest factor display is off.

**POWER:QUALITY:DISPLAY:VRMS**

Sets or returns the display state for the rms voltage readout.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

POWER:QUALITY:DISPLAY:VRMS {OFF|ON|0|1}

POWER:QUALITY:DISPLAY:VRMS?

**Arguments**

OFF or 0 turns off the rms voltage display.

ON or 1 turns on the rms voltage display.

**Examples**

POWER:QUALITY:DISPLAY:VRMS0 turns off the rms voltage display.

POWER:QUALITY:DISPLAY:VRMS? might return :POWER:QUALITY:DISPLAY:VRMS 1 indicating that the rms voltage display is on.

**POWER:QUALITY:FREQREFERENCE**

Sets or returns the power quality frequency reference.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

POWER:QUALITY:FREQREFERENCE {VOLTage|CURRent}

POWER:QUALITY:FREQREFERENCE?

**Arguments**

VOLTage sets voltage as the power quality frequency reference source.

CURRent sets current as the power quality frequency reference source.
Examples  

POWER:QUALITY:FREQREFERENCE CURRENT sets current as the power quality frequency reference source.

POWER:QUALITY:FREQREFERENCE? might return :POWER:QUALITY:FREQREFERENCE VOLTAGE indicating that the power quality frequency reference source is set to Voltage.

POWer:QUALity:FREQuency? (Query Only)

Returns the frequency measurement.

Conditions  

This command requires a DPO4PWR application module.

Group  

Power

Syntax  

POWER:QUALITY:FREQuency?

Examples  

POWER:QUALITY:FREQUENCY? might return :POWER:QUALITY:FREQUENCY 60 indicating that the frequency is set to 60 Hz.

POWer:QUALity:ICRESTfactor? (Query Only)

Returns the current crest factor measurement.

Conditions  

This command requires a DPO4PWR application module.

Group  

Power

Syntax  

POWER:QUALITY:ICRESTfactor?

Examples  

POWER:QUALITY:ICRESTFACTOR? might return :POWER:QUALITY:ICRESTfactor 1.4 indicating that the current crest factor value is set to 1.4.

POWer:QUALity:IRMS? (Query Only)

Returns the RMS current measurement.
**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
`POWer:QUALity:IRMS?`

**Examples**
`POWER:QUALITY:IRMS?` might return 
`:POWER:QUALITY:IRMS 1.00` indicating that the RMS current value is set to 1.00 A.

---

**POWER:QUALity:PHASEangle? (Query Only)**

Returns the phase angle measurement.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
`POWer:QUALity:PHASEangle?`

**Examples**
`POWER:QUALITY:PHASEANGLE?` might return 
`:POWER:QUALITY:PHASEANGLE 0` indicating that the phase angle is set to 0.

---

**POWER:QUALity:POWERFACTOR? (Query Only)**

Returns the power factor measurement.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
`POWer:QUALity:POWERFACTOR?`

**Examples**
`POWER:QUALITY:POWERFACTOR?` might return 
`:POWER:QUALITY:POWERFACTOR 1.0` indicating that the power factor is set to 1.0.
POWeR:QUALity:REACTpwr? (Query Only)

Returns the reactive power measurement.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWeR:QUALity:REACTpwr?

**Examples**
POWeR:QUALity:REACTpwr? might return:POWeR:QUALity:REACTpwr 100 indicating that the reactor power value is set to 100 VAR.

POWeR:QUALity:TRUEpwr? (Query Only)

Returns the true power measurement.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWeR:QUALity:TRUEpwr?

**Examples**
POWeR:QUALity:TRUEpwr? might return:POWeR:QUALity:TRUEpwr 1000 W indicating that the true power value is set to 1000 W.

POWeR:QUALity:VRMS? (Query Only)

Returns the RMS voltage measurement.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWeR:QUALity:VRMS?
Examples  
POWER:QUALITY:VRMS? might return:POWER:QUALITY:VRMS 115 indicating that the rms voltage value is set to 115 V.

POWer:REFLevel:ABSolute (No Query Form)

Sets the reference levels to their default unit values.

Conditions  
This command requires a DPO4PWR application module.

Group  
Power

Syntax  
POWer:REFLevel:ABSolute {SETTODEFaults}

Arguments  
SETTODEFaults sets the reference levels to their default values.

POWer:REFLevel:ABSolute:HIGH

Sets or returns the top reference level for power measurements.

Conditions  
This command requires a DPO4PWR application module.

Group  
Power

Syntax  
POWer:REFLevel:ABSolute:HIGH <NR3>; Ranges={D,-1e6,+1E6}
POWer:REFLevel:ABSolute:HIGH?

Arguments  
<NR3> is the absolute high value in volts. Default value is 0.0E+0

Examples  
POWER:REFLEVEL:ABSOLUTE:HIGH 2 sets the absolute high value to 2.000.

POWer:REFLevel:ABSolute:LOW

Sets or returns the low reference level for power measurements.
**Commands Listed in Alphabetical Order**

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
```
POWer:REFLevel:ABSolute:LOW <NR3>; Ranges={{D,-1e6,+1E6}}
POWer:REFLevel:ABSolute:LOW?
```

**Arguments**
- `<NR3>` is the absolute low value in volts. Default value is 0.0E+0.

**Examples**
```
POWER:REFLEVEL:ABSOLUTE:LOW 1.0 sets the absolute high value to 1.0000.
POWER:REFLEVEL:ABSOLUTE:LOW? might return
POWER:REFLEVEL:ABSOLUTE:LOW 1.0000 indicating that the absolute low value is set to 1 V.
```

**POWer:REFLevel:ABSolute:MID<1-3>**
Sets or returns the mid reference level for measurements. MID3 is specific to the power application.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
```
POWer:REFLevel:ABSolute:MID<1-3> <NR3>; Ranges={{D,-1e6,+1E6}}
POWer:REFLevel:ABSolute:MID<1-3>?
```

**Arguments**
- `<NR3>` is the absolute mid reference value.

**Examples**
```
POWER:REFLEVEL:ABSOLUTE:MID1 1 sets the mid reference voltage to 1V.
POWER:REFLEVEL:ABSOLUTE:MID1? might return
POWER:REFLEVEL:ABSOLUTE:MID1 0.0E+0 indicating that the absolute reference voltage level is set to 0.0E+0 V.
```

**POWer:REFLevel:HYSTeresis**
Sets or returns the measurement reference level hysteresis value.
**POWer:REFLevel:HYSTeresis**

This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**

```
POWer:REFLevel:HYSTeresis <NR3>
POWer:REFLevel:HYSTeresis?
```

**Arguments**

<NR3> is the hysteresis value.

**Examples**

```
POWER:REFLEVEL:HYSTERESIS 30
```

sets the hysteresis value to 30.

```
POWER:REFLEVEL:HYSTERESIS?
```

might return

```
POWER:REFLEVEL:HYSTERESIS 10
```

indicating that the hysteresis value is set to 10.

**POWer:REFLevel:METHod**

Sets or returns the method used to calculate the 0% and 100% reference level.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**

```
POWer:REFLevel:METHod {ABSolute|PERCent}
POWer:REFLevel:METHod?
```

**Arguments**

ABSolute specifies that the reference levels are set explicitly using the MEASUrement:REFlevel:ABSolute commands. This method is useful when precise values are required.

PERCent specifies that the reference levels are calculated as a percent of the signal amplitude. The percentages are defined using the MEASUrement:REFlevel:PERCent commands.

**Examples**

```
POWER:REFLEVEL:METHOD ABSOLUTE
```

specifies that explicit user-defined values are used for the reference levels.

```
POWER:REFLEVEL:METHOD? 
```

might return

```
POWER:REFLEVEL:METHOD PERCENT
```

indicating that the reference level units used are calculated as a percent relative to HIGH and LOW.
POWer:REFLevel:PERCent (No Query Form)

Sets the reference levels to the default percentage values.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:REFLevel:PERCent <SETTODEFaults>

**Arguments**
SETTODEFaults sets the reference levels to their default percentage values.

POWer:REFLevel:PERCent:HIGH

Sets or returns the top reference percent level for power measurements.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:REFLevel:PERCent:HIGH <NR3>; Ranges={D,0.0,100.0}
POWer:REFLevel:PERCent:HIGH?

**Arguments**
<NR3> is the high value in percent.

**Examples**
POWER:REFLEVEL:PERCENT:HIGH 95 sets the high reference level to 95% of the signal amplitude.

POWER:REFLEVEL:PERCENT:HIGH? might return :POWER:REFLEVEL:PERCENT:HIGH 90 indicating that the percentage high reference level is set to 90% of the signal amplitude.

POWer:REFLevel:PERCent:LOW

Sets or returns the low reference percent level for power measurements.

**Conditions**
This command requires a DPO4PWR application module.
Group: Power

Syntax: 
- `POWer:REFLevel:PERCent:LOW <NR3>; Ranges={D,0.0,100.0}`
- `POWer:REFLevel:PERCent:LOW?`

Arguments: `<NR3>` is the low value in percentage.

Examples: 
- `POWER:REFLEVEL:PERCENT:LOW 15` sets the low reference level to 15% of the signal amplitude.
- `POWER:REFLEVEL:PERCENT:LOW?` might return `:POWER:REFLEVEL:PERCENT:LOW 10` indicating that the percentage low reference level is set to 90% of the signal amplitude.

**POWER:REFLevel:PERCent:MID<1-3>**

Sets or returns the mid reference percent level for waveform measurements. Mid3 is specific to the power application.

Conditions: 
This command requires a DPO4PWR application module.

Syntax: 
- `POWer:REFLevel:PERCent:MID<1-3> <NR3>; Ranges={D,0.0,100.0}`
- `POWer:REFLevel:PERCent:MID<1-3>?`

Arguments: `<NR3>` is the mid value in percentage.

Examples: 
- `POWER:REFLEVEL:PERCENT:MID1 25` sets the mid reference voltage to 25%.
- `POWER:REFLEVEL:PERCENT:MID1?` might return `POWER:REFLEVEL:PERCENT:MID1 50.0000` indicating that the mid reference voltage level is set to 50%.

**POWER:RIPPle (No Query Form)**

Does a vertical autoset for ripple measurements or sets the vertical offset to 0.

Conditions: 
This command requires a DPO4PWR application module.
Group  Power

Syntax  POWER:RIPPLE {VERTAUTOset|VERTDEFault}

Arguments  VERTAUTOset automatically scales the source waveform to optimize ripple measurements.

          VERTDEFault sets the vertical offset of the source waveform to 0 volts (for voltage source) or 0 amperes (for current source).

POWER:RIPPLE:RESULTS:AMPLITUDE? (Query Only)

Returns the peak-to-peak ripple measurement.

Conditions  This command requires a DPO4PWR application module.

Group  Power

Syntax  POWER:RIPPLE:RESULTS:AMPLITUDE?

Examples  POWER:RIPPLE:RESULTS:AMPLITUDE? might return
          :POWER:RIPPLE:RESULTS:AMPLITUDE 1 indicating that the peak-to-peak amplitude is set to 1 V.

POWER:RIPPLE:RESULTS:MAX? (Query Only)

Returns the maximum of the peak-to-peak ripple measurements.

Conditions  This command requires a DPO4PWR application module.

Group  Power

Syntax  POWER:RIPPLE:RESULTS:MAX?

Examples  POWER:RIPPLE:RESULTS:MAX? might return :POWER:RIPPLE:RESULTS:MAX 1.1 indicating that the maximum peak-to-peak amplitude is set to 1.1 V.
POWer:RIPple:RESults:MEAN? (Query Only)

Returns the mean of the peak-to-peak ripple measurements.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:RIPple:RESults:MEAN?

**Examples**
POWer:RIPple:RESults:MEAN? might return
:POWer:RIPple:RESults:MEAN 1.0 indicating that the mean peak-to-peak amplitude is set to 1.0 V.

POWer:RIPple:RESults:MIN? (Query Only)

Returns the minimum of the peak-to-peak ripple measurement.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:RIPple:RESults:MIN?

**Examples**
POWer:RIPple:RESults:MIN? might return :POWer:RIPple:RESults:MIN 0.9 indicating that the minimum peak-to-peak amplitude is set to 0.9 V.

POWer:RIPple:RESults:STDdev? (Query Only)

Returns the standard deviation of the peak-to-peak ripple measurements.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:RIPple:RESults:STDdev?
Examples  
POWER:RIPPLE:RESULTS:STDDEV? might return :POWER:RIPPLE:RESULTS:STDDEV 0.14 indicating that the standard deviation of the peak-to-peak amplitude is set to 0.14.

POWer:RIPlPe:SOUrce

Sets or returns the source waveform for ripple tests. The voltage source waveform is specified using the POWer:VOLLageSource command and the current waveform is specified using the POWer:CURRentSoUrce command.

Conditions  
This command requires a DPO4PWR application module.

Group  
Power

Syntax  
POWer:RIPlPe:SOUrce {VOLLage|CURRent}

Arguments  
VOLLage specifies voltage source waveform for ripple tests.  
CURRent specifies current source waveform for ripple tests.

Examples  
POWER:RIPPLE:SOURCE CURRent sets the ripple source to Current.  
POWER:RIPPLE:SOURCE? might return :POWER:RIPPLE:SOURCE VOLL indicating that the ripple source is set to Voltage.

POWer:SOA:LINear:XMAX

Sets or returns the user XMAX value for use in linear SOA calculations.

Conditions  
This command requires a DPO4PWR application module.

Group  
Power

Syntax  
POWer:SOA:LINear:XMAX <NR3>  
POWer:SOA:LINear:XMAX?

Arguments  
<NR3> is the XMAX value used for linear SOA calculations.
POWer:SOA:LINear:XMIN

Sets or returns the user XMIN value for use in linear SOA calculations.

Conditions
This command requires a DPO4PWR application module.

Group
Power

Syntax
POWer:SOA:LINear:XMIN <NR3>
POWer:SOA:LINear:XMIN?

Arguments
<NR3> is the XMIN value used for linear SOA calculations.

Examples
POWer:SOA:LINear:XMIN 2 sets the XMIN value to 2.0000.
POWer:SOA:LINear:XMIN? might return:POWer:SOA:LINear:XMIN 0.0E+0 indicating that the XMIN value for linear plot type is set to 0.0E+0.

POWer:SOA:LINear:YMAX

Sets or returns the user YMAX value for use in linear SOA calculations.

Conditions
This command requires a DPO4PWR application module.

Group
Power

Syntax
POWer:SOA:LINear:YMAX <NR3>
POWer:SOA:LINear:YMAX?

Arguments
<NR3> is the YMAX value used for linear SOA calculations.
**Examples**

POWER:SOA:LINEAR:YMAX 700 sets the YMAX value to 700.0000.

POWER:SOA:LINEAR:YMAX ? might return :POWER:SOA:LINEar:YMAX 50.0000 indicating that the YMAX value for linear plot type is set to 50.0000.

**POWER:SOA:LINear:YMIN**

Sets or returns the user YMIN value for use in linear SOA calculations.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

POWER:SOA:LINear:YMIN <NR3>

POWER:SOA:LINear:YMIN?

**Arguments**

<NR3> is the YMIN value used for linear SOA calculations.

**Examples**

POWER:SOA:LINEAR:YMIN 2 sets the YMIN value to 2.0000.

POWER:SOA:LINEAR:YMIN ? might return :POWER:SOA:LINEar:YMIN 0.0E+0 indicating that the YMIN value for linear plot type is set to 0.0E+0.

**POWER:SOA:LOG:XMAX**

Sets or returns the user XMAX value for use in Log SOA calculations.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

POWER:SOA:LOG:XMAX <NR3>

POWER:SOA:LOG:XMAX?

**Arguments**

<NR3> is the XMAX value used for log SOA calculations.
**Examples**

```
POWER:SOA:LOG:XMAX 1 sets the XMAX value to 1.0000.

indicating that the XMAX value for log plot type is set to 1.0000E+3.
```

**POWER:SOA:LOG:XMIN**

Sets or returns the user XMIN value for use in Log SOA calculations.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

```
POWER:SOA:LOG:XMIN <NR3>
POWER:SOA:LOG:XMIN?
```

**Arguments**

<NR3> is the XMIN value used for log SOA calculations.

**Examples**

```
POWER:SOA:LOG:XMIN 2 sets the XMIN value to 2.0000.

indicating that the XMIN value for log plot type is set to 100.0000E-3.
```

**POWER:SOA:LOG:YMAX**

Sets or returns the user YMAX value for use in Log SOA calculations.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

```
POWER:SOA:LOG:YMAX <NR3>
POWER:SOA:LOG:YMAX?
```

**Arguments**

<NR3> is the YMAX value used for log SOA calculations.
Examples

POWER:SOA:LOG:YMAX 10 sets the YMAX value to 10.0000.

POWER:SOA:LOG:YMAX ? might return :POWER:SOA:LOG:YMAX 100.0000 indicating that the YMAX value for log plot type is set to 100.0000.

POWer:SOA:LOG:YMIN

Sets or returns the user YMIN value for use in Log SOA calculations.

Conditions

This command requires a DPO4PWR application module.

Group

Power

Syntax

POWer:SOA:LOG:YMIN <NR3>
POWer:SOA:LOG:YMIN?

Arguments

<NR3> is the YMIN value used for log SOA calculations.

Examples

POWER:SOA:LOG:YMIN 1 sets the YMIN value to 1.0000.

POWER:SOA:LOG:YMIN ? might return :POWER:SOA:LOG:YMIN 100.0000E-3 indicating that the YMIN value for log plot type is set to 100.0000E-3.

POWer:SOA:MASK:DEFine

Sets or returns the X (Volts) and Y (Amps) coordinates of the current SOA mask. You can specify the number of points from 2 to 10, minimum being 2. Successive X values must be ≥ the preceding X values. The number of XY points sent determines the value of NR_PT.

Conditions

This command requires a DPO4PWR application module.

Group

Power

Syntax

POWer:SOA:MASK:DEFine <NR3>
POWer:SOA:MASK:DEFine?

Arguments

<NR3> represents SOA mask coordinates.
Examples

```
POWER:SOA:MASK:DEFINE 10 specifies the SOA mask coordinates as 0.0E+0,0.0E+0,0.0E+0,0.0E+0,0.0E+0,30.0000,30.0000,30.0000,30.0000,30.0000,30.0000,2.5000,30.0000,2.5000.
```

```
POWER:SOA:MASK:DEFINE ? might return :POWER:SOA:MASK:DEFINE 0.0E+0,30.0000,25.0000,30.0000,30.0000,30.0000,2.5000,30.0000,0.0E+0 indicating that the SOA mask coordinates are set to 0.0E+0,30.0000,25.0000,30.0000,30.0000,30.0000,2.5000,30.0000,0.0E+0.
```

**POWer:SOA:MASK:MAXAmps**

Sets or returns the maximum current applied to SOA mask testing.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

```
POWer:SOA:MASK:MAXAmps <NR3>
POWer:SOA:MASK:MAXAmps?
```

**Arguments**

<NR3> is the maximum current applied to SOA mask testing.

**Examples**

```
POWER:SOA:MASK:MAXAMPS 40 sets the SOA mask testing current to 40 A.
```

```
POWER:SOA:MASK:MAXAMPS ? might return :POWER:SOA:MASK:MAXAMPS 30.0000 indicating that the SOA mask testing maximum current is set to 30 A.
```

**POWer:SOA:MASK:MAXVolts**

Sets or returns the maximum voltage applied to SOA mask testing.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

```
POWer:SOA:MASK:MAXVolts <NR3>
POWer:SOA:MASK:MAXVolts?
```

**Arguments**

<NR3> is the maximum voltage applied to SOA mask testing.
Examples  

POWER:SOA:MASK:MAXVOLTS 200 sets the SOA mask testing voltage to 200 V.  

POWER:SOA:MASK:MAXVOLTS? might return :POWER:SOA:MASK:MAXVOLTS 300.0000 indicating that the SOA mask testing maximum voltage is set to 300 V.

POWeR:SOA:MASK:MAXWatts

Sets or returns the maximum power applied to SOA mask testing.

Conditions  

This command requires a DPO4PWR application module.

Group  

Power

Syntax  

POWeR:SOA:MASK:MAXWatts <NR3>  
POWeR:SOA:MASK:MAXWatts?

Arguments  

<NR3> is the maximum power applied to SOA mask testing.

Examples  

POWER:SOA:MASK:MAXWATTS 1000 sets the SOA mask testing power to 1.0000E+3 W.  

POWER:SOA:MASK:MAXWATTS? might return :POWER:SOA:MASK:MAXWATTS 750.0000 indicating that the SOA mask testing maximum power is set to 750 W.

POWeR:SOA:MASK:NR_PT? (Query Only)

Returns the number of mask points defined.

Conditions  

This command requires a DPO4PWR application module.

Group  

Power

Syntax  

POWeR:SOA:MASK:NR_PT?

Examples  

**POWer:SOA:MASK:STATE**

Sets or returns the state of the mask for SOA calculations.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**

- `POWer:SOA:MASK:STATE {OFF|LIMITS|POINTS}`
- `POWer:SOA:MASK:STATE?`

**Arguments**

- **OFF**
  - Disables mask testing.

- **LIMITS**

- **POINTS**
  - Enables mask testing based on masks points defined.

**Examples**

- `POWer:SOA:MASK:STATE POINTS` enables mask testing power based on points defined.
- `POWer:SOA:MASK:STATE?` might return `:POWER:SOA:MASK:STATE LIMITS` indicating that the mask testing is enabled based on limits.

**POWer:SOA:MASK:STOPOnviol**

Sets or returns the enabled state of the mask stop on violation condition.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**

- `POWer:SOA:MASK:STOPOnviol {OFF|ON|0|1}`
- `POWer:SOA:MASK:STOPOnviol?`

**Arguments**

- **OFF** or 0 enables mask stop on violations.
- **ON** or 1 disables mask stop on violations.
Examples

```
POWER:SOA:MASK:STOPONVIOL 1 disables the state of the mask stop on violation condition.

POWER:SOA:MASK:STOPONVIOL ? might return :POWER:SOA:MASK:STOPONVIOL 0 indicating that the mask stop is enabled.
```

**POWer:SOA:PLOTTYPe**

Sets or returns the Safe Operating Area (SOA) plot type.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

```
POWer:SOA:PLOTTYPe {LOG|LINear}
POWer:SOA:PLOTTYPe?
```

**Arguments**

LOG for logarithmic SOA plot type.
LINear for linear SOA plot type.

**Examples**

```
POWER:SOA:PLOTTYPE LINear sets the SOA plot type to Linear.
POWER:SOA:PLOTTYPE ? might return :POWER:SOA:PLOTTYPE LOG indicating that the SOA plot type is set to Log.
```

**POWer:SOA:RESult:FAILures:QTY? (Query Only)**

Returns the number of failures in the test.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

```
POWer:SOA:RESULT:FAILures:QTY?
```

**Examples**

```
POWER:SOA:RESULT:FAILURES:QTY? might return :POWER:SOA:RESULT:FAILURES 0 indicating that the number of failures is 0.
```
**POWer:SOA:RESult:NUMACq? (Query Only)**

Returns the number of acquisitions in the test.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:SOA:RESult:NUMACq?

**Examples**
POWer:SOA:RESULT:NUMACQ? might return :POWer:SOA:RESULT:NUMACQ 10 indicating that the number of acquisitions is 10.

**POWer:SOA:RESult:STATe? (Query Only)**

Returns the pass/fail state of the SOA test.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:SOA:RESult:STATe?

**Returns**
PASS or FAIL.

**Examples**
POWer:SOA:RESULT:STATE? might return :POWer:SOA:MASK:STATE PASS indicating that the SOA test state is PASS.

**POWer:STATIstics (No Query Form)**

Clears all the accumulated statistics of all measurements. Performs the same function as the MEASUrement:STATIstics command.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power
Syntax  POWER:STATIsitics {RESET}

Arguments  RESET clears the measurement statistics.

POWer:STATIstics:MODE

Enables or disables the display of the measurement statistics. Performs the same function as the MEASUrement:STATIsitics:MODE command.

Conditions  This command requires a DPO4PWR application module.

Group  Power

Syntax  POWER:STATIsitics:MODE {OFF|ALL}

POWER:STATIsitics:MODE?

Arguments  ALL turns on measurement statistics display.

OFF turns all measurements statistics off.

Examples  POWER:STATIsitics:MODE OFF turns measurements statistics display off.

POWER:STATIsitics:MODE? might return :MEASUrement:STATIsitics:MODE ALL indicating that measurement statistics are turned on and all statistics are being displayed for each measurement.

POWer:STATIstics:WEIghting

Sets the number of samples which are included for the statistics computations for mean and the standard deviation. Performs the same function as the MEASUrement:STATIsitics:WEIghting command.

Conditions  This command requires a DPO4PWR application module.

Group  Power

Syntax  POWER:STATIsitics:WEIghting <NR1>;Ranges {L,2,1000}

POWER:STATIsitics:WEIghting?
Arguments  

<NR1> is the number of samples used for the mean and standard deviation statistical accumulations.

Examples  

POWER:STATISTICS:WEIGHTING 4 sets statistical weighting to four samples.

POWER:STATISTICS:WEIGHTING ? might return :POWER:STATISTICS:WEIGHTING 4 indicating that statistics weighting is currently set to 4 samples.

POWer:SWLoss:CONDCALCmethod  
Sets or returns the power application switching loss conduction calculation method.

Conditions  
This command requires a DPO4PWR application module.

Group  
Power

Syntax  
POWer:SWLoss:CONDCALCmethod {VOLTage|RDSon|VCEsat}
POWer:SWLoss:CONDCALCmethod?

Arguments  
VOLTage sets voltage as the conduction calculation method.
RDSon sets RDSon as the conduction calculation method.
VCEsat sets VCEsat as the conduction calculation method.

Examples  
POWER:SWLOSS:CONDCALCMETHOD RDSon sets the conduction calculation method to RDSon.

POWER:SWLOSS:CONDCALCMETHOD ? might return :POWER:SWLOSS:CONDCALCMETHOD VOLT indicating that the conduction calculation method is set to Voltage.

POWer:SWLoss:CONDUction:ENERGY:MAX? (Query Only)  
Returns the maximum conduction energy for switching loss calculation.

Conditions  
This command requires a DPO4PWR application module.

Group  
Power
Syntax  POWER:SWLoss:CONduction:ENERGY:MAX?

:POWER:SWLoss:CONduction:ENERGY:MAX 1 indicating that the maximum conduction energy switching loss calculation is set to 1 J.

POWER:SWLoss:CONduction:ENERGY:MEAN? (Query Only)

Returns the mean conduction energy in the conduction period for the switching loss calculation.

Conditions  This command requires a DPO4PWR application module.

Group  Power

Syntax  POWER:SWLoss:CONduction:ENERGY:MEAN?

:POWER:SWLoss:CONduction:ENERGY:MEAN 1 indicating that the mean conduction energy switching loss calculation is set to 1 J.

POWER:SWLoss:CONduction:ENERGY:MIN? (Query Only)

Returns the minimum conduction energy for the switching loss calculation.

Conditions  This command requires a DPO4PWR application module.

Group  Power

Syntax  POWER:SWLoss:CONduction:ENERGY:MIN?

:POWER:SWLoss:CONduction:ENERGY:MIN 1 indicating that the minimum conduction energy switching loss calculation is set to 1 J.
POWer:SWLoss:CONDuction:POWER:MAX? (Query Only)

Returns the maximum conduction power for the switching loss calculation.

Conditions
This command requires a DPO4PWR application module.

Group
Power

Syntax
POWer:SWLoss:CONDuction:POWER:MAX?

Examples
POWER:SWLOSS:CONDUCTION:POWER:MAX? might return
:POWER:SWLOSS:CONDUCTION:POWER:MAX 1 indicating that the
maximum conduction power switching loss calculation is set to 1 W.

POWer:SWLoss:CONDuction:POWER:MEAN? (Query Only)

Returns the mean conduction power for the switching loss calculation.

Conditions
This command requires a DPO4PWR application module.

Group
Power

Syntax
POWer:SwLoss:CONDuction:POWER:MEAN?

Examples
:POWER:SWLOSS:CONDUCTION:POWER:MEAN 1 indicating that the
mean conduction power switching loss calculation is set to 1 W.

POWer:SWLoss:CONDuction:POWER:MIN? (Query Only)

Returns the minimum conduction power for the switching loss calculation.

Conditions
This command requires a DPO4PWR application module.

Group
Power

Syntax
POWer:SwLoss:CONDuction:POWER:MIN?
Commands Listed in Alphabetical Order

Examples

POWER:SWLOSS:CONDUCTION:POWER:MIN? might return
:POWER:SWLOSS:CONDUCTION:POWER:MIN 1 indicating that the
minimum conduction power switching loss calculation is set to 1 W.

POWer:SWLoss:DIsply

Sets or returns the display selection for switching loss results: All measurements,
energy loss measurements or power loss measurements.

Conditions

This command requires a DPO4PWR application module.

Group

Power

Syntax

POWer:SWLoss:DIsply {ALL|ENERGYLoss|POWERLoss}
POWer:SWLoss:DIsply?

Arguments

ALL displays both energy and power loss measurements in the results.
ENERGYLoss displays only energy loss measurements in the results.
POWERLoss displays only power loss measurements in the results.

Examples

POWER:SWLOSS:DISPLAY ENERGYLoss displays only energy loss measurements in the results.
POWER:SWLOSS:DISPLAY? might return :POWER:SWLOSS:DISPLAY ALL indicating that the results displays both energy and power loss measurements.

POWer:SWLoss:GATe:POLarity

Sets or returns the switching loss gate polarity.

Conditions

This command requires a DPO4PWR application module.

Group

Power

Syntax

POWer:SWLoss:GATe:POLarity {FALL|RISe}
POWer:SWLoss:GATe:POLarity?

MSO4000 and DPO4000 Series Programmer Manual 2-313
Arguments

- **FALL** sets falling edge as the switching loss gate polarity.
- **RISe** sets rising edge as the switching loss gate polarity.

Examples

- `POWER:SWLOSS:GATE:POLARITY FALL` sets the gate polarity to Fall.

**POWER:SWLoss:GATE:TURNON**

Sets or returns the gate turn on level for switching loss power measurements.

Conditions

This command requires a DPO4PWR application module.

Group

Power

Syntax

- `POWER:SWLoss:GATE:TURNON <NR3>`
- `POWER:SWLoss:GATE:TURNON?`

Examples

- `POWER:SWLOSS:GATE:TURNON 3.5` sets the gate turn on level to 3.5000.
- `POWER:SWLOSS:GATE:TURNON`? might return `POWER:SWLOSS:GATE:TURNON 1.5` indicating that the gating turn on level is set to 1.5.

**POWER:SWLoss:NUMCYCles? (Query Only)**

Returns the number of cycles counted for the switching loss calculation.

Conditions

This command requires a DPO4PWR application module.

Group

Power

Syntax

- `POWER:SWLoss:NUMCYCles? <NR3>`

Examples

- `POWER:SWLoss:NUMCYCles?` might return the number of cycles for switching loss calculation, if the `POWER:TYPE` is set to Switching Loss (`POWER:TYPE:SWITCHingLoss`). If the query times out with the Switching
Loss power enabled, it indicates the possibility of incorrect settings and might return the following:

*ESR?
20
ALLEV?
2202,"Measurement error, No period found;
:power:swloss:numcycles?",410,"Query INTERRUPTED"

**POWer:SWLoss:RDSon**

Sets or returns the user RDSON value for use in switching loss calculations when the conduction calculation method is RDSON.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**

```
POWer:SWLoss:RDSon <NR3>
POWer:SWLoss:RDSon?
```

**Arguments**

<NR3> is the RDSON switching loss calculation.

**Examples**

```
POWer:SWLOSS:RDSon 30 sets the RDSon value to 30.0000.
POWer:SWLOSS:RDSon ? might return :POWer:SWLOSS:RDSon 20.0000E-3 indicating that the RDSon value is set to 20.0000E-3.
```

**POWer:SWLoss:REFLevel:ABSoolute:GATEMid**

Sets or returns the mid voltage reference level used in switching loss power measurements in volts.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**

```
POWer:SWLoss:REFLevel:ABSolute:GATEMid <NR3>
POWer:SWLoss:REFLevel:ABSolute:GATEMid?
```

MSO4000 and DPO4000 Series Programmer Manual 2-315
Arguments  

<NR3> is the mid voltage reference level in volts.

Examples  

POWER:SWLOSS:REFLEVEL:ABSOLUTE:GATEMID 100 sets the absolute gate mid reference voltage to 100.0000 V.

POWER:SWLOSS:REFLEVEL:ABSOLUTE:GATEMID ? might return :POWER:SWLOSS:REFLEVEL:ABSOLUTE:GATEMID 50.0000 indicating that the gate mid reference voltage is set to 50.0000 V.

POWeR:SWLoss:REFLevel:ABSolute:LOWCurrent  

Sets or returns the low current reference level used in switching loss power measurements in amperes.

Conditions  

This command requires a DPO4PWR application module.

Group  

Power

Syntax  

POWeR:SwLoss:REFLevel:ABSolute:LOWCurrent <NR3>

POWeR:SwLoss:REFLevel:ABSolute:LOWCurrent?

Arguments  

<NR3> is the low voltage current level in amperes.

Examples  

POWER:SWLOSS:REFLEVEL:ABSOLUTE:LOWCURRENT 2.8 sets the absolute reference low current to 2.8000 A.

POWER:SWLOSS:REFLEVEL:ABSOLUTE:LOWCURRENT ? might return :POWER:SWLOSS:REFLEVEL:ABSOLUTE:LOWCURRENT 5.0000 indicating that the absolute reference low current is set to 5.0000 A.

POWeR:SWLoss:REFLevel:ABSolute:LOWVoltage  

Sets or returns the low voltage reference level used in switching loss power measurements in volts.

Conditions  

This command requires a DPO4PWR application module.

Group  

Power
Syntax
POWer:SWLoss:REFLevel:ABSolute:LOWVoltage <NR3>
POWer:SWLoss:REFLevel:ABSolute:LOWVoltage?

Arguments
<NR3> is the low voltage reference level in volts.

Examples
POWer:SWLoss:REFLevel:ABSolute:LOWVoltage 2.5 sets the absolute reference low voltage to 2.5000 V.

POWer:SWLoss:REFLevel:ABSolute:LOWVoltage ? might return :POWER:SWLoss:REFLevel:ABSolute:LOWVoltage 5.0000 indicating that the absolute reference low voltage is set to 5.0000 V.

POWer:SWLoss:REFLevel:PERCent:GATEMid

Sets or returns the mid voltage reference level used in switching loss power measurements in percent.

Conditions
This command requires a DPO4PWR application module.

Group
Power

Syntax
POWer:SWLoss:REFLevel:PERCent:GATEMid <NR3>
POWer:SWLoss:REFLevel:PERCent:GATEMid?

Arguments
<NR3> is the mid voltage reference level in volts.

Examples
POWer:SWLoss:REFLevel:PERCent:GATEMid 30 sets the gate mid reference voltage to 30%.

POWer:SWLoss:REFLevel:PERCent:GATEMid ? might return :POWER:SWLoss:REFLevel:PERCent:GATEMid 50.0000 indicating that the gate mid reference voltage is set to 50%.

POWer:SWLoss:REFLevel:PERCent:LOWCurrent

Sets or returns the low current reference level used in switching loss power measurements in percent.

Conditions
This command requires a DPO4PWR application module.
**Commands Listed in Alphabetical Order**

<table>
<thead>
<tr>
<th>Group</th>
<th>Power</th>
</tr>
</thead>
</table>

**Syntax**

- `POWer:SWLoss:REFLevel:PERCent:LOWCurrent <NR3>`
- `POWer:SWLoss:REFLevel:PERCent:LOWCurrent?`

**Arguments**

- `<NR3>` is the low voltage reference level percent.

**Examples**

- `POWER:SWLOSS:REFLEVEL:PERCENT:LOWCURRENT 15` sets the reference low current to 15%.

**POWer:SWLoss:REFLevel:PERCent:LOWVoltage**

Sets or returns the low voltage reference level used in switching loss power measurements in percent.

**Conditions**

This command requires a DPO4PWR application module.

<table>
<thead>
<tr>
<th>Group</th>
<th>Power</th>
</tr>
</thead>
</table>

**Syntax**

- `POWer:SWLoss:REFLevel:PERCent:LOWVoltage <NR3>`
- `POWer:SWLoss:REFLevel:PERCent:LOWVoltage?`

**Arguments**

- `<NR3>` is the low voltage reference level in percent.

**Examples**

- `POWER:SWLOSS:REFLEVEL:PERCENT:LOWVOLTAGE 10` sets the reference low voltage to 10%.
- `POWER:SWLOSS:REFLEVEL:PERCENT:LOWVOLTAGE?` might return `:POWER:SWLOSS:REFLEVEL:PERCENT:LOWVOLTAGE 5.0000` indicating that the reference low voltage is set to 5%.

**POWer:SWLoss:TOFF:ENERGY:MAX? (Query Only)**

Returns the maximum Toff energy for the switching loss calculation.

**Conditions**

This command requires a DPO4PWR application module.
**Group**  
Power

**Syntax**  
`POWER:SWLoss:TOFF:ENERGY:MAX?`

**Examples**  
`:POWER:SWLoss:TON:ENERGY:MAX 1` indicating that the maximum Toff energy switching loss calculation is set to 1 J.

Returns the mean Toff energy for the switching loss calculation.

**Conditions**  
This command requires a DPO4PWR application module.

**Group**  
Power

**Syntax**  
`POWER:SWLoss:TOFF:ENERGY:MEAN?`

**Examples**  
`:POWER:SWLoss:TON:ENERGY:MEAN 1` indicating that the mean Toff energy switching loss calculation is set to 1 J.

Returns the minimum Toff energy for the switching loss calculation.

**Conditions**  
This command requires a DPO4PWR application module.

**Group**  
Power

**Syntax**  
`POWER:SWLoss:TOFF:ENERGY:MIN?`

**Examples**  
`:POWER:SWLoss:TON:ENERGY:MIN 1` indicating that the minimum Toff energy switching loss calculation is set to 1 J.
POWeR:SWLoss:TOFF:POWER:MAX? (Query Only)

Returns the maximum Toff power for the switching loss calculation.

Conditions
This command requires a DPO4PWR application module.

Group
Power

Syntax
POWeR:SWLoss:TOFF:POWER:MAX?

Examples
POWeR:SWLoss:TOFF:POWER:MAX? might return
:POWeR:SWLoss:TON:POWER:MAX 1 indicating that the maximum Toff power
switching loss calculation is set to 1 W.

POWeR:SWLoss:TOFF:POWER:MEAN? (Query Only)

Returns the mean Toff power for the switching loss calculation.

Conditions
This command requires a DPO4PWR application module.

Group
Power

Syntax
POWeR:SWLoss:TOFF:POWER:MEAN?

Examples
POWeR:SWLoss:TOFF:POWER:MEAN? might return
:POWeR:SWLoss:TON:POWER:MEAN 1 indicating that the mean Toff power
switching loss calculation is set to 1 W.

POWeR:SWLoss:TOFF:POWER:MIN? (Query Only)

Returns the minimum Toff power for the switching loss calculation.

Conditions
This command requires a DPO4PWR application module.

Group
Power

Syntax
POWeR:SWLoss:TOFF:POWER:MIN?
Examples

```
POWER:SWLOSS:TOFF:POWER:MIN?  might return
:POWER:SWLOSS:TON:POWER:MIN  1 indicating that the minimum Toff power
switching loss calculation is set to 1 W.
```


Returns the maximum Ton energy for the switching loss calculation.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

```
POWe:r:SWLoss:TON:ENERGY:MAX?
```

**Examples**

```
:POWe:r:SWLoss:TON:ENERGY:MAX  1 indicating that the maximum Ton energy
switching loss calculation is set to 1 J.
```

**POWe:r:SWLoss:TON:ENERGY:MEAN? (Query Only)**

Returns the mean Ton energy for the switching loss calculation.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

```
POWe:r:SWLoss:TON:ENERGY:MEAN?
```

**Examples**

```
:POWe:r:SWLoss:TON:ENERGY:MEAN  1 indicating that the mean Ton energy
switching loss calculation is set to 1 J.
```

**POWe:r:SWLoss:TON:ENERGY:MIN? (Query Only)**

Returns the minimum Ton energy for the switching loss calculation.

**Conditions**

This command requires a DPO4PWR application module.
Commands Listed in Alphabetical Order

**Group** Power

**Syntax** `POWer:SWLoss:TON:ENERGY:MIN?`

**Examples** `POWer:SWLoss:TON:ENERGY:MIN?` might return `:POWer:SWLoss:TON:ENERGY:MIN 1` indicating that the minimum Ton energy switching loss calculation is set to 1 J.


Returns the maximum Ton power for the switching loss calculation.

**Conditions** This command requires a DPO4PWR application module.

**Group** Power

**Syntax** `POWer:SWLoss:TON:POWER:MAX?`

**Examples** `POWer:SWLoss:TON:POWER:MAX?` might return `:POWer:SWLoss:TON:POWER:MAX 1` indicating that the maximum Ton power switching loss calculation is set to 1 W.


Returns the mean Ton power for the switching loss calculation.

**Conditions** This command requires a DPO4PWR application module.

**Group** Power

**Syntax** `POWer:SwLoss:TON:POWER:MEAN?`

**Examples** `POWer:SwLoss:TON:POWER:MEAN?` might return `:POWer:SwLoss:TON:POWER:MEAN 1` indicating that the mean Ton power switching loss calculation is set to 1 W.
POWer:SWLoss:TON:POWER:MIN? (Query Only)

Returns the minimum Ton power for the switching loss calculation.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:SWLoss:TON:POWER:MIN?

**Examples**
POWer:SWLoss:TON:POWER:MIN? might return
:POWER:SWLoss:TON:POWER:MIN 1 indicating that the minimum Ton power
switching loss calculation is set to 1 W.

POWer:SWLoss:TOTal:ENERGY:MAX? (Query Only)

Returns the maximum total energy for the switching loss calculation.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:SWLoss:TOTal:ENERGY:MAX?

**Examples**
POWer:SWLoss:TOTAL:ENERGY:MAX? might return
:POWER:SWLoss:TOTAL:ENERGY:MAX 1 indicating that the maximum
conduction energy switching loss calculation is set to 1 J.

POWer:SWLoss:TOTal:ENERGY:MEAN? (Query Only)

Returns the mean total energy for the switching loss calculation.

**Conditions**
This command requires a DPO4PWR application module.

**Group**
Power

**Syntax**
POWer:SWLoss:TOTal:ENERGY:MEAN?
Examples  

PO\textit{W}er:SW\textit{Loss}:TOT\textit{al}:\textit{ENERGY}:\textit{MEAN}? might return  
\textit{PO\textit{W}er:SW\textit{Loss}:TOT\textit{al}:\textit{ENERGY}:\textit{MEAN} 1} indicating that the mean conduction  
energy switching loss calculation is set to 1 J.

**PO\textit{W}er:SW\textit{Loss}:TOT\textit{al}:\textit{ENERGY}:\textit{MIN}? (Query Only)**

Returns the minimum total energy for the switching loss calculation.

**Conditions**  
This command requires a DPO4PWR application module.

**Group**  
Power

**Syntax**  
\textit{PO\textit{W}er:SW\textit{Loss}:TOT\textit{al}:\textit{ENERGY}:\textit{MIN}?}

**Examples**  
PO\textit{W}er:SW\textit{Loss}:TOT\textit{al}:\textit{ENERGY}:\textit{MIN}? might return  
\textit{PO\textit{W}er:SW\textit{Loss}:TOT\textit{al}:\textit{ENERGY}:\textit{MIN} 1} indicating that the minimum  
conduction energy switching loss calculation is set to 1 J.

**PO\textit{W}er:SW\textit{Loss}:TOT\textit{al}:\textit{POWER}:\textit{MAX}? (Query Only)**

Returns the maximum total power loss.

**Conditions**  
This command requires a DPO4PWR application module.

**Group**  
Power

**Syntax**  
\textit{PO\textit{W}er:SW\textit{Loss}:TOT\textit{al}:\textit{POWER}:\textit{MAX}?}

**Examples**  
PO\textit{W}er:SW\textit{Loss}:TOT\textit{al}:\textit{POWER}:\textit{MAX}? might return  
\textit{PO\textit{W}er:SW\textit{Loss}:TOT\textit{al}:\textit{POWER}:\textit{MAX} 1} indicating that the maximum total  
power switching loss calculation is set to 1 W.

**PO\textit{W}er:SW\textit{Loss}:TOT\textit{al}:\textit{POWER}:\textit{MEAN}? (Query Only)**

Returns the mean total power loss.

**Conditions**  
This command requires a DPO4PWR application module.
Commands Listed in Alphabetical Order

**Group**  
Power

**Syntax**  
`POWER:SWLoss:TOTAL:POWER:MEAN?`

**Examples**  
`POWER:SWLoss:TOTAL:POWER:MEAN?` might return  
`:POWER:SWLoss:TOTAL:POWER:MEAN 1` indicating that the mean total power switching loss calculation is set to 1 W.

**POWER:SWLoss:TOTAL:POWER:MIN? (Query Only)**

Returns the minimum total power loss.

**Conditions**  
This command requires a DPO4PWR application module.

**Group**  
Power

**Syntax**  
`POWER:SWLoss:TOTAL:POWER:MIN?`

**Examples**  
`POWER:SWLoss:TOTAL:POWER:MIN?` might return  
`:POWER:SWLoss:TOTAL:POWER:MIN 1` indicating that the minimum total power switching loss calculation is set to 1 W.

**POWER:SWLoss:VCEsat**

Sets or returns VCESAT value for use in switching loss calculations when the conduction calculation method is VCESAT.

**Conditions**  
This command requires a DPO4PWR application module.

**Group**  
Power

**Syntax**  
`POWER:SWLoss:VCEsat <NR3>`

`POWER:SWLoss:VCEsat?`

**Arguments**  
`<NR3>` is the VCESat switching loss calculation.
Examples

POWER:SWLOSS:VCESAT 5 sets the VCEsat value to 5.0000.

POWER:SWLOSS:VCESAT? might return :POWER:SWLOSS:VCESAT 2.0000 indicating that the VCEsat value is set to 2.0000.

POWer:TYPE

Sets or returns the power application measurement type.

Conditions

This command requires a DPO4PWR application module.

Group

Power

Syntax

POWer:TYPE {NONE|QUALity|SWITCHingloss|SOA|HARMonics|RIPPle|MODULationanalysis|DESKew}

POWer:TYPE?

Arguments

NONE Use to set the measurement type to None.

QUALity Use the power quality functions to obtain measurements and statistics about the general power quality in your test circuit

SWITCHingloss Use the switching loss functions to obtain the power loss and energy loss across the acquired waveform, including turn-on loss, turn-off loss, conduction loss, and total loss. Typically, use these functions to characterize losses in power supply switching devices, as they switch on and off.

SOA Use the safe operating functions to obtain an X-Y display of the switching device-under-test’s voltage and current. Also use them to perform a mask test of the X-Y signal relative to the graphical X-Y description of the device specification table. The safe operating area is typically the voltage and current values that a semiconductor can operate without damaging itself.

HARMonics Use the harmonics functions to obtain the frequency spectrum of the source waveform and associated measurement values. Harmonic measurements can help one perform in-depth troubleshooting of power quality problems.

RIPPle Use the ripple functions to obtain measurements and statistics for the AC components of the acquired waveform. Ripples are often found on top of a large DC signal.

MODULationanalysis Use the modulation functions to obtain a trend plot of a measurement value across the acquired waveform. This is useful for showing the variations in the modulated switching signal.

DESKew Run the deskew procedure to match the delays through the probes. Different probes introduce different delays between the probe tip and the
oscilloscope. Many oscilloscope users do not have to worry about this because they use the same type of probe on all channels. Power measurement users, however, frequently use both a voltage probe and a current probe. A current probe typically has a larger delay than a voltage probe, so setting deskew values becomes important.

**Examples**

`POWER:TYPE HARMonics` sets the measurement type to Harmonics.

`POWER:TYPE?` might return `POWER:TYPE NON` indicating that no measurement type is set.

**POWer:VOLTAGESOurce**

Sets or returns the voltage source for the power application.

**Conditions**

This command requires a DPO4PWR application module.

**Group**

Power

**Syntax**

```
POWer:VOLTAGESOurce {CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4}
POWer:VOLTAGESOurce?
```

**Arguments**

`CH1–CH4` or `REF1–REF4` sets channel or ref as the voltage source.

**Examples**

`POWER:VOLTAGESOURCE CH1` sets the voltage source as CH1.

`POWER:VOLTAGESOURC?` might return `POWER:VOLTAGESOURC CH1` indicating that the voltage source is set to CH1.

**nPSC**

Sets or returns 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
Commands Listed in Alphabetical Order

**Syntax**

```
*PSC {OFF|ON|NR1>}
*PSC?
```

**Related Commands**

DESE, *ESE, FACTory, *RST, *SRE

**Arguments**

OFF sets the power-on status clear flag to false.

ON sets the power-on status clear flag to true.

<NR1> = 0 sets the power-on status clear flag to false. This disables the power-on clear allowing the oscilloscope 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 preventing 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**

Sets or returns 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?
```

**Related Commands**

PASSWord

**Arguments**

<Block> is a block containing up to 300 ASCII characters.

<QString> is a string containing up to 300 ASCII characters.

**Examples**

*PUD #229This oscilloscope belongs to me stores the string "This oscilloscope belongs to me" in the user protected data area.

*PUD? might return #221PROPERTY OF COMPANY X
**RCL (No Query Form)**

This command restores the state of the oscilloscope from a copy of the settings stored in memory (The settings are stored using the *SAV command).

**Group**  
Save and Recall

**Syntax**  
*RCL  <NR1>*

**Related Commands**  

**Arguments**  
<NR1> is a value in the range from 1 to 10, which specifies a saved setup storage location.

**Examples**  
*RCL 3* restores the oscilloscope from a copy of the settings stored in memory location 3.

**RECALL:SETUp (No Query Form)**

Restores the state of the oscilloscope from a copy of the settings stored in memory. The settings are stored using the *SAV command.

**Group**  
Save and Recall

**Syntax**  
RECALL:SETUp {FACTory|<NR1>|<file path>}

**Related Commands**  

**Arguments**  
FACTory restores the factory setup.

<NR1> is a value in the range from 1 to 10, which specifies a saved setup storage location.

<file path> specifies a location for an oscilloscope setup file. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<filename>.<extension> and one or <dir>s are optional. If you do not specify them, the oscilloscope will read the file from the default directory (see FILESystem:CWD). <filename> stands for a filename; the use of wildcard characters in filenames is not supported. Filename extensions are not required, but highly recommended.
RECALL:SETUP FACTORY recalls (and makes current) the oscilloscope setup to its factory defaults.

RECALL:SETUP 2 recalls the oscilloscope setup from setup storage location 2.

RECALL:SETUP "TEK00000.SET" recalls the setup from the file TEK00000.SET in the current working directory.

RECALL:WAVEform (No Query Form)

This command (no query form) recalls a stored waveform to a reference location.

Group
Save and Recall

Syntax
RECALL:WAVEform <file path>,REF<x>

Related Commands
SAVE:WAVEform, FILESystem:CWD, FILESystem?

Arguments
REF<x> specifies a location in internal reference memory. Reference memory location values range from 1 through 4.

<file path> specifies a location for an oscilloscope setup file. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<filename>.<extension> and one or <dir>s are optional. If you do not specify them, the oscilloscope will read the file from the default directory (see FILESystem:CWD). <filename> stands for a filename; the use of wildcard characters in filenames is not supported. Filename extensions are not required, but highly recommended.

Examples
RECALL:WAVEFORM "TEK00000.ISF",REF1 recalls the waveform stored in the file named TEK00000.ISF from the current directory for waveforms to the reference location 1.

REF<x>?(Query Only)

Returns reference waveform data for the channel specified by <x>, where x is the reference channel number.

Group
Vertical

Syntax
REF<x>?
**REF<x>:DATE? (Query Only)**

Returns the date that reference waveform data for channel <x> was copied into the internal reference memory, where x is the reference channel number.

**Group** Vertical

**Syntax**

`REF<x>:DATE?`

**REF<x>:HORizontal:DELay:TIMe**

Sets or returns the horizontal delay time for reference waveform <x>, where x is the reference channel number. The delay time is expressed in seconds and is limited to ±5 times the reference horizontal scale.

**Group** Vertical

**Syntax**

`REF<x>:HORizontal:DELay:TIMe <NR3>`

`REF<x>:HORizontal:DELay:TIMe?`

**Arguments**

<NR3> is the delay time in seconds.

**Examples**

`REF2:HORIZONTAL:DELAY:TIME 4.0E-6` sets the horizontal delay time for the REF2 waveform to 4 μs.

**REF<x>:HORizontal:SCAle**

Sets or returns the horizontal scale for reference waveform <x>, where x is the reference channel number.

**Group** Vertical

**Syntax**

`REF<x>:HORizontal:SCAle <NR3>`

`REF<x>:HORizontal:SCAle?`

**Arguments**

<NR3> is the horizontal scale in seconds.

**Examples**

REF<x>:LABel

Sets or returns the reference waveform label for the channel specified by <x>, where x is the reference channel number.

**Group**  Vertical

**Syntax**  
REF<x>:LABel <Qstring>
REF<x>:LABel?

**Arguments**  
<Qstring> is an alphanumeric string of text, enclosed in quotes, that contains the label text for the reference channel <x> waveform. The text string is limited to 30 characters.

**Examples**  

REF<x>:TIME? (Query Only)

Returns the time that reference waveform data was copied into the internal reference memory for reference channel <x>, where x is the reference channel number.

**Group**  Vertical

**Syntax**  
REF<x>:TIME?

**Examples**  
REF4:TIME? might return “16:54:05”.

REF<x>:VERTical:POSition

Sets or returns the vertical position of the reference waveform specified by <x>, where x is the reference channel number.

Increasing the position value of a waveform causes the waveform to move up, and decreasing the position value causes the waveform to move down. Position adjusts only the display position of a waveform. The position value determines the vertical graticule coordinate at which signal values are displayed. For example, if the position for Reference 3 is set to 2.0, the signal represented by that reference will be displayed at 2.0 divisions above the center of the screen.
Commands Listed in Alphabetical Order

Group  Vertical

Syntax  REF<x>:VERTical:POSition <NR3>
        REF<x>:VERTical:POSition?

Related Commands  CH<x>:POSition, MATH[1]:VERTical:POSition

Arguments  <NR3> is the desired position, in divisions from the center horizontal graticule. The range is from -5.0 to 5.0 divisions.

Examples  REF2:VERTICAL:POSITION 1.3E+00 positions the Reference 2 input signal 1.3 divisions above the center horizontal graticule.

REF1:VERTICAL:POSITION? might return :REF1:VERTICAL:POSITION -1.3000E+00 indicating that the current position of Reference 1 is 1.3 divisions below the center horizontal graticule.

REF<x>:VERTical:SCAle

Sets or returns the vertical scale for the reference waveform specified by <x>, where x is the reference channel number.

Each waveform has a vertical scale parameter. For a signal with constant amplitude, increasing the Scale causes the waveform to be displayed smaller. Decreasing the scale causes the waveform to be displayed larger.

Scale affects all waveforms, but affects reference and math waveforms differently from channel waveforms:

- For reference and math waveforms, this setting controls the display only, graphically scaling these waveforms and having no affect on the acquisition hardware.
- For channel waveforms, this setting controls the vertical size of the acquisition window as well as the display scale. The range and resolution of scale values depends on the probe attached and any other external factors you have specified.

Group  Vertical

Syntax  REF<x>:VERTical:SCAle <NR3>
        REF<x>:VERTical:SCAle?

Related Commands  CH<x>:SCAle, MATH[1]:VERTical:SCAle
Commands Listed in Alphabetical Order

**Arguments**

<NR3> is the gain in user units-per-division.

**Examples**

REF4:VERTICAL:SCALE 100E-03 sets the Reference 4 scale to 100 mV per division.

REF4:VERTICAL:SCALE? might return :REF2:VERTICAL:SCALE 1.0000e+00 indicating that the current vertical scale setting for Reference 2 is 1 V per division.

**REM (No Query Form)**

Embeds a comment within programs as a means of internally documenting the programs. The oscilloscope 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 oscilloscope will ignore.

**RST (No Query Form)**

Resets the oscilloscope to the factory default settings. The *RST command does not alter the following:

- Calibration data that affect device specifications
- The Output Queue
- The Service Request Enable Register setting
- The Power-on status clear flag setting
- Alias definitions
- Stored settings
- The *PUD? Response
- Any of the values associated with the DATA command.
- Oscilloscope password
Commands Listed in Alphabetical Order

**Group**
Status and Error

**Syntax**
* RST

**Related Commands**

**Arguments**
None

**Examples**
* RST resets the oscilloscope settings to factory defaults.

**SAV (No Query Form)**
Stores the state of the oscilloscope to a specified memory location. You can use the *RCL command to restore the oscilloscope to this saved state at a later time.

**Group**
Save and Recall

**Syntax**
*SAV <NR1>

**Related Commands**
*RCL, RECALL:SETUp, SAVe:SETUp

**Arguments**
<NR1> specifies a location in which to save the state of the oscilloscope. Location values range from 1 through 10. Using an out-of-range location value causes an execution error. Any settings that have been stored previously at this location will be overwritten.

**Examples**
*SAV 2 saves the current oscilloscope state in memory location 2.

**SAVe:ASSIgn:TYPe**
Sets or returns the assignment of the data to be saved when the front-panel Save button is pressed.

**Group**
Save and Recall

**Syntax**
SAVe:ASSIgn:TYPe {IMAGE|WAVEform|SETUp}
SAVe:ASSIgn:TYPe?
Commands Listed in Alphabetical Order

**Arguments**

**IMAGe** assigns the Save button to save screen images.

**WAVEform** assigns the Save button to save waveforms.

**SETUp** assigns the Save button to save setups.

**SAVe:EVENTtable:BUS<x> (No Query Form)**

Saves the data from bus<x> to a specified file and location; where x is the bus number.

**Group**

Save and Recall

**Syntax**

```
SAVe:EVENTtable:BUS<x> <file path>
```

**Arguments**

<file path> is a quoted string that defines the file name and path location where the event table will be stored.

**NOTE.** *<filename>* stands for a filename of up to 125 characters, followed by a period (".") and the three-character extension. Waveform files should have a .csv extension for comma-separated spreadsheet format files.

**SAVe:IMAGe (No Query Form)**

Saves a capture of the screen image into the specified file. Supported image formats are PNG, Windows Bitmap, and TIFF. If an extension for a supported file type is added to the file name, then the corresponding format will be used. If no supported extension is added to the file, the format to use will be determined by the value obtained from the :SAVe:IMAGe:FILEFormat? query.

**Group**

Save and Recall

**Syntax**

```
SAVe:IMAGe <file path>
```

**Related Commands**

SAVe:ASSIgn:TYPe

**Arguments**

<file path> is a filename, including path, where the image will be saved. If you do not specify a directory, the oscilloscope will store the file in the current working directory. File name extensions are not required but are highly...
Commands Listed in Alphabetical Order

The path “D:/foo.png” will save an image to the file “foo.png” on the Compact Flash card.

**SAVe:IMAGe:FILEFormat**

Sets or returns the file format to use for saving screen images.

**NOTE.** The file format is not automatically determined by the file name extension. You need to choose a file format with an extension which is consistent with the selected file format.

**Group**  
Save and Recall

**Syntax**  
SAVe:IMAGe:FILEFormat {PNG|BMP|TIff}

SAVe:IMAGe:FILEFormat?

**Related Commands**  
SAVe:IMAGe

**Arguments**
- **PNG** saves the file in Portable Network Graphics format.
- **BMP** saves the file in Microsoft Windows bitmap format.
- **TIff** saves the file in Tagged Image File Format.

**SAVe:IMAGe:INKSaver**

Sets or returns the current ink saver setting for the SAVe:IMAGe command. If set to “ON” or “1”, images will be generated using the ink saver palette. If set to “OFF” or “0”, images will be generated using the standard palette.

**Group**  
Save and Recall

**Syntax**  
SAVe:IMAGe:INKSaver {OFF|ON|0|1}

SAVe:IMAGe:INKSaver?

**Related Commands**  
SAVe:IMAGe

**Arguments**
- **OFF** or 0 generates images from the Inksaver palette.
- **ON** or 1 generates images using the Standard palette.
SAVe:IMAGE:LAYout

Sets or returns the layout to use for saved screen images.

Group  
Save and Recall

Syntax  
SAVe:IMAGE:LAYout {LANDscape|PORTRait}
SAVe:IMAGE:LAYout?

Arguments  
LANDscape specifies that screen images are saved in landscape format.
PORTRait specifies that screen images are saved in portrait format.

SAVe:SETUp (No Query Form)

Stores the state of the oscilloscope to a specified memory location. You can later use the *RCL command to restore the oscilloscope to this saved state.

Group  
Save and Recall

Syntax  
SAVe:SETUp {<file path>|<NR1>}

Related Commands  
*RCL, RECALL:SETUp, *SAV

Arguments  
<file path> is the target location for storing the setup file. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<filename>. <extension> and one or <dir>s are optional. If you do not specify them, the oscilloscope will store the file in the current working directory. <filename> stands for a filename. (Use of wildcard characters in filenames is not supported.) Filename extensions are not required but are highly recommended. For setups, use the extension ".SET".

<NR1> specifies a location for saving the current front-panel setup. The front-panel setup value ranges from 1 to 10. Using an out-of-range value causes an execution error. Any settings that have been stored previously at this location will be overwritten.

Examples  
SAVe:SETUp 5 saves the current oscilloscope setup in memory location 5.
SAVe:SETUp "TEK00000.SET" saves the current oscilloscope setup in the file TEK00000.SET in the current working directory.
SAVe:WAVEform (No Query Form)

This command saves a specified waveform or all displayed waveforms (excluding serial bus waveforms). Only individual analog waveforms (CH<x>, MATH and REF<x>) can be saved to reference memory locations.

You can save all displayed waveforms, excluding serial bus waveforms, to a single CSV file when the SAVE:WAVEFORM:FILEFORMAT is set to SPREADSHEET.

You can save all displayed waveforms, excluding serial bus waveforms and digital channels (MSO models) to consecutive ISF (internal save format) files when the SAVE:WAVEFORM:FILEFORMAT is set to INTERNAL.

**Group**
Save and Recall

**Syntax**
```
SAVe:WAVEform [<wfm>,{REF<x>}] | [<wfm>, <QString>] | [ALL, <QString>]
```

**Related Commands**
RECAll:WA VEform, SA Ve:WA VEform:FILEFormat

**Arguments**
- `<wfm>, <REF<x>>` saves the specified waveform to the specified reference memory location. `<wfm>` can be any live analog channel (where `<x>` is the channel number), the MATH1 waveform, or another reference waveform (such as REF1).
- `<wfm>, <QString>` saves the specified waveform to the file specified in the quoted string argument. Any live channel (such as CH1), the MATH1 waveform, any reference waveform, and, for MSO models, digital waveforms (such as D0) can be saved to a file. Digital waveforms can only be saved when the SAVE:WAVEFORM:FILEFORMAT is set to SPREADSHEET.
- `ALL, <QString>` saves all displayed waveforms, excluding serial bus waveforms, to a single CSV file specified by the quoted string argument when the SAVE:WAVEFORM:FILEFORMAT is set to SPREADSHEET, or saves all displayed waveforms, excluding serial bus waveforms and digital channel waveforms (MSO models), to individual ISF (internal save format) files with a file name prefix specified by the argument with an underscore (_) and the waveform ID (such as CH1, REF1, MATH) appended to the file name(s).

**Examples**
- `SAVE:WAVEFORM CH1,REF1` saves the CH1 waveform in reference memory location 1.
- `:SAVE:WAVEFORM:FILEFORMAT SPREADSHEET; :SAVE:WAVEFORM ALL, "D:/test_folder/test1_all.csv"` saves all displayed waveforms (excluding serial bus waveforms) to `D:/test_folder/test1_all.csv`. 
SAVE:WAVEform:FILEFormat

Specifies or returns the file format for saved waveforms. Waveform header and timing information is included in the resulting file of non-internal formats. The oscilloscope saves DPO waveforms as a 500 x 200 matrix, with the first row corresponding to the most recently acquired data. The values specified by DATa:STARt and DATa:STOP determine the range of waveform data to output. In the event that DATa:STOP value is greater than the current record length, the current record length determines the last output value.

Group Save and Recall

Syntax

SAVE:WAVEform:FILEFormat {INTERNal|SPREADSheet}
SAVE:WAVEform:FILEFormat?

Related Commands CURVe, DATa, DATa:STARt, DATa:STOP, SAVE:WAVEform, WFMInpre:NR_Pt, WFMOutpre:NR_Pt?

Arguments

INTERNal specifies that waveforms are saved in an internal format, using a .isf filename extension. These files can be recalled as reference waveforms. When this argument is specified, the settings specified via the DATa:STARt and DATa:STOP commands have no meaning as the entire waveform is saved.

SPREADSheet specifies that waveform data is saved in a format that contains comma delimited values. These waveform data files are named using the .csv filename extension. Saving waveforms in CSV format enables spreadsheet programs to import the data.

Examples

SAVE:WAVEFORM:FILEFORMAT INTERNAL specifies that the internal file format is the format used for saving waveforms.

SAVE:WAVEFORM:FILEFORMAT? might return :SAVE:WAVEFORM:FILEFORMAT INTERNAL indicating that waveforms are saved using the internal format.

SAVE:WAVEform:GATIng

Specifies whether save waveform operations should save the entire waveform (NONE) or a specified portion of the waveform.
**Save and Recall**

**Syntax**

```
SAVE:WAVEform:GATIng {NONE|CURSors|SCREEN}
```

**Arguments**

- **CURSors** turns on cursors and the gates are the waveform record points at the cursor positions.
- **NONE** saves the entire waveform.
- **SCREEN**, if zoom is on, the gates are the start and end waveform record points of the zoom (upper) graticule, otherwise the gates are the start and end waveform record points of the main graticule.

**Examples**

```
SAVE:WAVEFORM:GATING CURSors
```

specifies that, when the waveform gating is set to cursors, save waveform operations should save the waveform points between the cursors. If cursors are turned off, waveform gating automatically reverts to NONE.

**Search**

**Syntax**

```
SEARCH?
```

**Examples**

```
SEARCH? might return:

```
SEARCH:SEARCH<x>:COPy (No Query Form)

Copies the search criteria to the trigger, or the trigger criteria to a search. <x> is the search number.

**Group**
Search

**Syntax**
SEARCH:SEARCH<x>:COPy {SEARCHtotrigger|TRIGgertosearch}

**Arguments**
SEARCHtotrigger copies the search criteria to the trigger
TRIGgertosearch copies the trigger criteria to the search
SEARCH:SEARCH<x>:STATE

Sets the search state to on or off. <x> is the search number, which is always 1. The query form returns the search state.

Group: Search

Syntax:
```
SEARCH:SEARCH<x>:STATE {<NR1>|OFF|ON}  
SEARCH:SEARCH<x>:STATE?
```

Arguments:
- OFF or <NR1> = 0 sets the search state to off.
- ON or <NR1> ≠ 0 sets the search state to on.

SEARCH:SEARCH<x>:TOTAL? (Query Only)

Returns the total number of matches for the search. The total number of matches may be than the number of marks placed. <x> is the search number, which is always 1.

Group: Search

Syntax:
```
SEARCH:SEARCH<x>:TOTAL?
```

Returns:
- <NR1> is the total number of matches.

SEARCH:SEARCH<x>:TRIGger:A:BUS? (Query Only)

Returns the serial search type. <x> is the search number, which is always 1. There are four serial buses, B1 through B4.

Conditions: This command requires a DPO4AUTO or DPO4EMBD application module.

Group: Search

Syntax:
```
SEARCH:SEARCH<x>:TRIGger:A:BUS?
```

Returns:
- I2C specifies the Inter-IC bus.
SPI specifies the Serial Peripheral Interface bus.

CAN specifies the Controller Area Network bus.

**Examples**

`SEARCH:SEARCH1:TRIGGER:A:BUS?` might return

```
```

**SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:AUDIO:CONDITION**

Sets or returns the search trigger condition for the AUDIO bus: Start of Frame or Data.

**Conditions**

This command requires DPO4AUDIO application module.

**Group**

Search

**Syntax**

```
{SOF|DATA}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:AUDIO:CONDITION?
```

**Arguments**

SOF enables triggering at the start of frame for the audio bus.

DATA enables triggering on matching data for the audio bus.


Sets or returns the search trigger data upper word for the AUDIO bus.

**Conditions**

This command requires DPO4AUDIO application module.

**Group**

Search

**Syntax**

```
<string>
```
Arguments <String> specifies the search data trigger data upper word.


Sets or returns the search trigger data offset for the AUDIO bus.

Conditions This command requires DPO4AUDIO application module.

Group Search

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:AUDio:DATa:OFFSet <NR1>

Arguments <NR1> is the search trigger data offset value.


Sets or returns the search trigger data qualifier for the AUDIO bus.

Conditions This command requires DPO4AUDIO application module.

Group Search

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:AUDio:DATa:QUALifier {LESSThan|MOREThan|EQUAL|UNEQual|LESSEQual|MORREEQual|INrange|OUTrange}

Arguments LESSThan sets the search trigger data qualifier to less than.
MOREThan sets the search trigger data qualifier to greater than.
EQUAL sets the search trigger data qualifier to equal.
UNEQual sets the search trigger data qualifier to not equal.
LESSEQual sets the search trigger data qualifier to less than or equal.
MORREEQual sets the search trigger data qualifier to greater than or equal.
INrange sets the search trigger data qualifier to in range.
OUTrange sets the search trigger data qualifier to out of range.

Sets or returns the search trigger data lower word for the AUDIO bus.

**Conditions**
This command requires DPO4AUDIO application module.

**Group**
Search

**Syntax**

**Arguments**
- `<String>` is the search trigger data lower word.


Sets or returns the search trigger data alignment for the AUDIO bus.

**Conditions**
This command requires DPO4AUDIO application module.

**Group**
Search

**Syntax**

**Arguments**
- `EITHER` aligns the search trigger data to either left or right.
- `LEFT` aligns the search trigger data to the left.
- `RIGHT` aligns the search trigger data to the right.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:CONDition

Sets or returns the search condition for a CAN trigger search. SEARCH<x> is the search number and B<x> is the bus number.

**Conditions**
This command requires a DPO4AUTO application module.
Group  
Search

Syntax  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:CONDition
{SOF|FRAMEtype|IDentifier|DATA|IDANDDATA|EOF|ACKMISS}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:CONDition?

Arguments  
SOF specifies a search based on the start of frame.
FRAMEtype specifies a search based on the frame type.
IDentifier specifies a search based on the frame identifier.
DATA specifies a search based on the frame data.
IDANDDATA specifies a search based on the frame identifier and data.
EOF specifies a search based on the end of frame.
ACKMISS specifies a search based on the missing ACK field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:DIRection

Sets or returns the CAN search to be valid for Read, Write, or Either condition if the criteria is IDentifier. SEARCH<x> is the search number and B<x> is the bus number. This only applies if the search condition is IDentifier.

Conditions  
This command requires a DPO4AUTO application module.

Group  
Search

Syntax  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:DIRection
{READ|WRITE|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:DIRection?

Related Commands  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:CONDition

Arguments  
READ specifies the read condition.
WRITE specifies the write condition.
NOCARE specifies either a read or write condition.
**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:QUALifier**

Sets or returns the CAN data qualifier for a search. *SEARCH<x>* is the search number and *B<x>* is the bus number. This only applies if the trigger condition is IDANDDATA or DATA.

**Conditions**

This command requires a DPO4AUTO application module.

**Group**

Search

**Syntax**

```plaintext
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:QUALifier
{LESSThan|Than|EQUal|UNEQual|LESSEQual|EQual}
```

**Related Commands**


**Arguments**

- **LESSThan** searches for bus data less than the value specified by `SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:VALue`.

- **Than** searches for bus data greater than the value specified by `SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:VALue`.


- **UNEQual** searches for bus data not equal to the value specified by `SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:VALue`.

- **LESSEQual** searches for bus data less equal to the value specified by `SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:VALue`.

- **EQual** searches for bus data equal to the value specified by `SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:VALue`.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:SIZe**

Sets or returns the length of the data string in bytes to be used for a CAN search if the search condition is DATA or IDANDDATA. *SEARCH<x>* is the search number and *B<x>* is the bus number.

**Conditions**

This command requires a DPO4AUTO application module.

**Group**

Search
Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:SIZE <NR1>

Arguments

<NR1> is the data string length in bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:VALue

Sets or returns the binary data string to be used for a CAN search if the search condition is ID or IDANDDATA. SEARCH<x> is the search number and B<x> is the bus number.

Conditions

This command requires a DPO4AUTO application module.

Group

Search

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:VALue <bin>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:VALue?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:QUALifier

Arguments

<bin> is the data in binary format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:FRAMEtype

Sets or returns the CAN Frame Type to be used if the trigger search condition is Frame Type. SEARCH<x> is the search number and B<x> is the bus number.

Conditions

This command requires a DPO4AUTO application module.

Group

Search

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:FRAMEtype {DATA|REMOTE|ERROR|OVERLoad}?
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:FRAMEtype?

Arguments

DATA specifies a data frame. REMote specifies a remote frame.
ERROR specifies an error frame.
OVERLoad specifies an overload frame.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN{:IDentifier|:ADDRess}:M0De**

Sets or returns the CAN addressing mode for a trigger search to a standard or extended format. SEARCH<x> is the search number and B<x> is the bus number.

**Conditions**
This command requires a DPO4AUTO application module.

**Group**
Search

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN{:IDentifier|:ADDRess}:M0De {STandard|EXTended}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN{:IDentifier|:ADDRess}:M0De?
```

**Arguments**

STandard specifies an 11-bit identifier field.
EXTended specifies a 29-bit identifier field.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN{:IDentifier|:ADDRess}:VALue**

Sets or returns the binary address string to be used for a CAN trigger search if the search condition is ID or IDANDDATA. SEARCH<x> is the search number and B<x> is the bus number.

**Conditions**
This command requires a DPO4AUTO application module.

**Group**
Search

**Syntax**
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN{:IDentifier|:ADDRess}:VALue <bin>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN{:IDentifier|:ADDRess}:VALue?
```

**Arguments**

<bin> is the address in binary format.
**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:CONDition**

Sets or returns the trigger condition for FLEXRAY.

**Group**  
Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:CONDition
{SOF|FRAMETypeid|CYCLEcount|HEADER|DATA|IDANDDATA|EOF|ERROR}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:CONDition?
```

**Arguments**

- **SOF** sets the FLEXRAY trigger condition to start of frame.
- **FRAMETypeid** sets the FLEXRAY trigger condition to a frame type id.
- **CYCLEcount** sets the FLEXRAY trigger condition to cycle count.
- **HEADER** sets the FLEXRAY trigger condition to header.
- **DATA** sets the FLEXRAY trigger condition to data.
- **IDANDDATA** sets the FLEXRAY trigger condition to ID and data.
- **EOF** sets the FLEXRAY trigger condition to EOF.
- **ERROR** sets the FLEXRAY trigger condition to an error.

**Examples**

```
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:CONDITION ERROR sets the FLEXRAY condition to ERROR
```

**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:HIVALue**

Sets or returns the binary data string to be used for FLEXRAY cycle count high value.

**Group**  
Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount: HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount: HIVALue?
```
Arguments

<QString> is a quoted string that is the cycle count high value.

Examples

“110010” sets the cycle count high value to 110010.

might return
"XXXXXX" indicating the cycle count high value is set to don’t cares.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:QUALifier

Sets or returns the FLEXRAY cycle count qualifier.

Group

Search

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:QUALifier
{LESSThan|MOREThan|EQUal|UNEQual|LESSEQual|MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:QUALifier?

Arguments

LESSThan sets the cycle count qualifier to less than.
MOREThan sets the cycle count qualifier to greater 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

LESSTHAN sets the cycle count qualifier to LESSTHAN.

might return
EQUAL indicating the cycle count qualifier is set to EQUAL.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:VALue

Sets or returns the binary data string to be used for FLEXRAY cycle count low value.

**Group**  
Search

**Syntax**
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:VALue?

**Arguments**
<QString> is a quoted string that is the binary cycle count value.

**Examples**
"110010" sets the cycle count value to 110010.
might return  
"XXXXXXXX" indicating the cycle count value is don’t cares.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:DATa:HIVALue

Sets or returns the high binary data string used for FLEXRAY trigger if trigger condition is ID or IDANDDATA.

**Group**  
Search

**Syntax**
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:DATa:HIVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:DATa:HIVALue?

**Arguments**
<QString> is a quoted string that is the binary high value.

**Examples**
"11001010" sets the high value 11001010.
might return  
"XXXXXXXX" indicating the high value is don’t cares.
**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:DATa:OFFSET**

Sets or returns the offset of the data string in bytes to be used for FLEXRAY trigger.

**Group**  
Search

**Syntax**


<NR1>


**Arguments**

<NR1> is the FLEXRAY data offset in bytes. A byte offset of -1 signifies don’t care, and no byte offset is used. The instrument will trigger or match any byte value that fits.

**Examples**


**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:DATa:QUALifier**

Sets or returns the FLEXRAY data qualifier.

**Group**  
Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:DATa:QUALifier

{LESSThan|MOREThan|EQUal|UNEQual|LESSEQual|MORTEEQual|INrange|OUTrange}

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:DATa:QUALifier?

**Arguments**

LESSThan sets the FLEXRAY data qualifier to less than.

MOREThan sets the FLEXRAY data qualifier to greater than.

EQUal sets the FLEXRAY data qualifier to equal.

UNEQual sets the FLEXRAY data qualifier to not equal.

LESSEQual sets the FLEXRAY data qualifier to less than or equal.

MORTEEQual sets the FLEXRAY data qualifier to greater than or equal.
**INrange** sets the FLEXRAY data qualifier to in range.

**OUTrange** sets the FLEXRAY data qualifier to out of range.

**Examples**

```
```

**LESSTHAN** sets the data qualifier to LESSTHAN.

```
```

might return

```
```

**EQUAL** indicating the data qualifier is EQUAL.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:DATa:SIZe**

Sets or returns the length of the data string in bytes to be used for FLEXRAY trigger.

**Group** Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:DATa:SIZe?
```

**Arguments**

<NR1> is the length of the data string in bytes. Length range is 1 to 8.

**Examples**

```
```

sets the data size to 8 bytes.

```
```

might return

```
```

indicating the data size is 1 byte.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:DATa:VALue**

Sets or returns the low binary data string used for FLEXRAY trigger condition if trigger condition is ID or IDANDDATA.

**Group** Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:DATa:VALue?
```

MSO4000 and DPO4000 Series Programmer Manual 2-355
Arguments

<qString> is a quoted string that is the low binary data string to be used for FLEXRAY trigger condition if trigger condition is ID or IDANDDATA.

Examples


"11001010" sets the binary data string to 11001010.


might return


"XXXXXXXX" indicates the binary data string is don’t cares.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:EOFTYPE

Sets or returns the end of frame type used for FLEXRAY trigger.

Group

Search

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:EOFTYPE

{STATic|DYNAMic|ANY}

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:EOFTYPE?

Arguments

STATic sets the FLEXRAY end of frame type to STATIC.

DYNAMic sets the FLEXRAY end of frame type to DYNAMIC.

ANY sets the FLEXRAY end of frame type to ANY type.

Examples


SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:ERRTYPE

Sets or returns the error type be used for FLEXRAY trigger.

Group

Search

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:ERRTYPE

{CRCHeader|CRCTrailer|SYNCFrame|STARTupnosync|NULLFRStatic|NULLFRDynamic}

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:ERRTYPE?
Commands Listed in Alphabetical Order

**Arguments**
- **CRCHeader** sets the FLEXRAY error type to CRC header.
- **CRCTrailer** sets the FLEXRAY error type to CRC trailer.
- **SYNCFrame** sets the FLEXRAY error type to SYNC frame.
- **STARTupnosync** sets the FLEXRAY error type to start up with no sync.
- **NULLFRStatic** sets the FLEXRAY error type to null frame static.
- **NULLFRDynamic** sets the FLEXRAY error type to null frame dynamic.

**Examples**
```
```
sets the FLEXRAY error type SYNCFRAME.

```
```
might return
```
```
indicating the FLEXRAY error type is CRCHEADER.

**SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:FLEXray:FRAMEID:HIVALue**

Sets or returns the binary data string used for FLEXRAY frame ID high value.

**Group**
Search

**Syntax**
```
```

```
<QString>
```

**Arguments**
- `<QString>` is a quoted string representing the binary data string used for FLEXRAY frame ID high value.

**Examples**
```
"00101100101"
```
sets the frame id high value to 00101100101.

```
```
might return
```
"XXXXXXXXXXX"
```
indicating the frame id high value is don’t cares.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:FRAMEID:QUALifier**

Sets or returns the FLEXRAY frame ID qualifier.
Group: Search

Syntax:

```plaintext
{LESSThan|MOREThan|EQUal|UNEQual|LESSEQual|MOREEQual|INrange|OUTrange}
```

Arguments:

- `LESSThan` sets the frame id qualifier to LESSThan
- `MOREThan` sets the frame id qualifier to MOREThan
- `QUal` sets the frame id qualifier to QUal
- `UNEQual` sets the frame id qualifier to UNEQual
- `LESSEQual` sets the frame id qualifier to LESSEQual
- `MOREEQual` sets the frame id qualifier to MOREEQual
- `INrange` sets the frame id qualifier to INrange
- `OUTrange` sets the frame id qualifier to OUTrange

Examples:


Sets or returns the binary data string to be used for FLEXRAY frame ID low value.

Group: Search

Syntax:

```plaintext
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:FRAMEID:VALue?
```

Arguments:

- `<QString>` is a quoted string representing the binary data string used for FLEXRAY frame ID low value.
Examples


**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:FRAMEType**

Sets or returns the frame type for FLEXRAY.

**Group**
Search

**Syntax**
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:FRAMEType
{NORMAL|PAYLOAD|NULL|SYNC|STARTup}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:FRAMEType?

**Arguments**
NORMAL sets the FLEXRAY frame type to normal.
PAYLOAD sets the FLEXRAY frame type to payload.
NULL sets the FLEXRAY frame type to NULL.
SYNC sets the FLEXRAY frame type to sync.
STARTup sets the FLEXRAY frame type to start up.

**Examples**

**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:HEADER:CRC**

Sets or returns the CRC portion of the binary header string to be used for FLEXRAY trigger.

**Group**
Search

**Syntax**
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:HEADER:CRC?
**Arguments**

<QString> is a quoted string representing the CRC portion of the binary header string used for FLEXRAY trigger.

**Examples**

"00110010101" sets the CRC portion of the binary header string 00110010101.

SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CRC "XXXXXXXXXX" indicating the CRC portion of the binary header string is don’t cares.


Sets or returns the cycle count portion of the binary header string used for FLEXRAY trigger.

**Group**

Search

**Syntax**


**Arguments**

<QString> is a quoted string representing the cycle count portion of the binary header string used for FLEXRAY trigger.

**Examples**

"001101" sets the cycle count to 001101.

might return


Sets or returns the frame id portion of the binary header string used for FLEXRAY trigger.

**Group**

Search
Syntax

<QString>

Arguments

<QString> is a quoted string representing the frame id portion of the binary header string used for FLEXRAY trigger.

Examples

"10110010101" sets the frame id portion of the binary header string to 10110010101.

"XXXXXXXXXXX" indicating the frame id portion of the binary header string is don’t cares.


Sets or returns the indicator bits portion of the binary header string used for FLEXRAY trigger.

Group
Search

Syntax

<QString>

Arguments

<QString> is a quoted string representing the indicator bits portion of the binary header string used for FLEXRAY trigger.

Examples

"10100" sets the indicator bits of the header to 10100.

"XXXXX" indicating the indicator bits of the header are don’t cares.


Sets or returns the payload length portion of the binary header string used for FLEXRAY trigger.
Commands Listed in Alphabetical Order


- **Syntax**: `SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:FLEXray:HEADER:PAYLength <QString>

- **Arguments**: `<QString>` is a quoted string representing the payload length portion of the binary header string used for FLEXRAY trigger.


**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:MODe**

- **Syntax**: `SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:MODe {ADDR7|ADDR10}
  SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:MODe?`

- **Arguments**: `ADDR7` specifies 7-bit addresses.
  `ADDR10` specifies 10-bit addresses.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:TYPe**

- **Syntax**: `SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:TYPe`

- **Conditions**: This command requires a DPO4EMBD application module.

- **Group**: Search

- **Syntax**: `SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:TYPe` sets or returns the I2C address type. `SEARCH<x>` is the search number and `B<x>` is the bus number.
**Conditions**

This command requires a DPO4EMBD application module.

**Group**

Search

**Syntax**

```
{GENeralcall|STARTbyte|HSmode|EEPROM|USER}
```

**Arguments**

- `GENeralcall` specifies the GENeralcall address type.
- `STARTbyte` specifies the STARtbyte address type.
- `HSmode` specifies the HSmode address type.
- `EEPROM` specifies the EEPROM address type.
- `USER` specifies a user address.

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:VALue <bin>
```

Sets or returns the binary address string to be used for an I2C trigger search if the search condition is ADDR or ADDRANDDATA. `SEARCH<x>` is the search number and `B<x>` is the bus number.

**Conditions**

This command requires a DPO4EMBD application module.

**Group**

Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:VALue <bin>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:VALue?
```

**Arguments**

- `<bin>` is the address in binary format.

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:CONDition
```

Sets or returns the search condition for an I2C trigger search. `SEARCH<x>` is the search number and `B<x>` is the bus number.

**Conditions**

This command requires a DPO4EMBD application module.

**Group**

Search
Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:CONDITION
{START|STOP|REPEATstart|ACKMISS|ADDRESS|DATA|ADDRANDDATA}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:CONDITION?

Arguments
START specifies a search based on a start condition.
STOP specifies a search based on a stop condition.
REPEATstart specifies a search based on a repeat of start condition.
ACKMISS specifies a search based on a missing acknowledgement condition.
ADDRESS specifies a search based on an address.
DATA specifies a search based on a data condition.
ADDRANDDATA specifies a search based on an address and data condition.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:DIRection
Sets or returns the I2C search condition to be valid on a Read, Write, or Either condition. SEARCH<x> is the search number and B<x> is the bus number.

Conditions
This command requires a DPO4EMBD application module.

Group
Search

Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:DIRection
{READ|WRITE|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:DIRection?

Arguments
READ specifies a read condition.
WRITE specifies a write condition.
NOCARE specifies either a read or write condition.

Sets or returns the length of the data string in bytes to be used for an I2C trigger search if the search condition is DATA or ADDRANDDATA. SEARCH<x> is the search number and B<x> is the bus number.

Conditions
This command requires a DPO4EMBD application module.
Commands Listed in Alphabetical Order

**Group** Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:SIZe <NR1>

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:SIZe?

**Arguments**

<NR1> is the data string length in bytes.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:VALue**

Sets or returns the binary data string to be used for an I2C trigger search if the search condition is DATA or ADDRANDDATA. SEARCH<x> is the search number and B<x> is the bus number.

**Conditions**

This command requires a DPO4EMBD application module.

**Group** Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:VALue <bin>

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:VALue?

**Arguments**

<bin> is the data in binary format.

**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:LIN:CONDition**

Sets or returns the search condition for a LIN search.

**Group** Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:LIN:CONDition {SYNCField|IDentifier|DATA|IDANDDATA|WAKEup|SLEEP|ERROR}

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:LIN:CONDition?

**Arguments**

SYNCField specifies to search on the sync field.

IDentifier specifies to search on the identifier.

DATA specifies to search on the data.

IDANDDDATA specifies to search on the identifier and the data.

WAKEup specifies to search on wake up.
SLEEP specifies to search on sleep.
ERROR specifies to search on errors.

**Examples**

**SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:LIN:DATA:HIVALue**
Sets or returns the binary data string to be used for LIN searches if the search condition is ID or IDANDDATA.

**Group** Search

**Syntax**

**Arguments**
<QString> is a quoted string of 1s, 0s, or Xs representing the binary data string to be used for LIN searches if the search condition is ID or IDANDDATA.

**Examples**
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX X XXXXXXXXXXXXX" indicating the high value is don’t care.

Sets or returns the LIN data qualifier. This only applies if the trigger condition is IDANDDATA or DATA.

**Group** Search

**Syntax**
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:LIN:DATA:QUALifier {LESSThan|MORThan|EQUal|UNEQual|LESSEqual|MOREEQual|INrange|OUTrange}
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.
- **MOREEQUAL** 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


**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:LIN:DATa:SIZe**

Sets or returns the length of the data string in bytes to be used for LIN Search, if search condition is DATA or IDANDDATA.

Group

Search

Syntax

- `SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:LIN:DATa:SIZe <NR1>`

Arguments

- `<NR1>` is the length of the data in bytes.

Examples


**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:LIN:DATa:VALue**

Sets or returns the binary data string used for a LIN search if the search condition is ID or IDANDDATA.
Commands Listed in Alphabetical Order

**Group** Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:LIN:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:LIN:DATa:VALue?

**Arguments**

<QString> is the binary data string for the search.

**Examples**


**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:LIN:ERRTYPE**

Sets or returns the error type used for a LIN Search.

**Group** Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:LIN:ERRTYPE {SYNC|PARity|CHecksum|HEADertime|RESPtime|FRAMetime}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:LIN:ERRTYPE?

**Arguments**

SYNC specifies a sync error type.
PARity specifies a parity error type.
Checksum specifies a checksum error type.
HEADertime specifies a header time error type.
RESPtime specifies a response time error type.
RAMetime specifies a frame time error type.

**Examples**


**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:LIN:IDenti fier:VALue**

Sets or returns the binary address string used for LIN search if search condition is ID or IDANDDATA.
Group Search


Arguments <QString> is a quoted string specifying the binary address string to be used for LIN search if search condition is ID or IDANDDATA.


**SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:PARALLEL:VALUE**
Sets or returns the binary data string to be used for a Parallel trigger search. SEARCH<x> is the search number and B<x> is the bus number.

Group Search

Syntax SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:PARALLEL:VALUE <QString>
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:PARALLEL:VALUE?

Arguments <QString> is the binary data string.

Sets or returns the condition for an RS-232 trigger search. SEARCH<x> is the search number and B<x> is the bus number.

Conditions This command requires a DPO4COMP application module.

Group Search

Arguments

- **RXSTArt** specifies a search based on the RX Start Bit.
- **RXDATA** specifies a search based on RX Data.
- **RXENDPacket** specifies a search based on the RX End of Packet condition.
- **TXSTArt** specifies a search base on the TX Start Bit.
- **TXDATA** specifies a search based on TX Data.
- **TXENDPacket** specifies a search based on the TX End of Packet condition.


Sets or returns the length of the data string for an RS-232 trigger search if the trigger condition is RX. **SEARCH<x>** is the search number and **B<x>** is the bus number.

**Conditions**

This command requires a DPO4COMP application module.

**Group**

Search

**Syntax**

```
```

**Arguments**

**<NR1>** is the length of the data string in Bytes.


Sets or returns the binary data string for an RS-232 trigger search if the condition involves RX. **SEARCH<x>** is the search number and **B<x>** is the bus number.

**Conditions**

This command requires a DPO4COMP application module.

**Group**

Search

**Syntax**

```
```

Sets or returns the length of the data string to be used for an RS-232 trigger search if the Trigger condition is TX. SEARCH<x> is the search number and B<x> is the bus number.

**Conditions**
This command requires a DPO4COMP application module.

**Group**
Search

**Syntax**

**Arguments**
<NRI> is the length of the data string in Bytes.


Sets or returns the binary data string to be used for an RS-232 trigger search if the condition involves RX. SEARCH<x> is the search number and B<x> is the bus number.

**Conditions**
This command requires a DPO4COMP application module.

**Group**
Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:CONDition

Sets or returns the search condition for a SPI trigger search. SEARCH<x> is the search number and B<x> is the bus number.

**Conditions**
This command requires a DPO4EMBD application module.

**Group**
Search
Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:CONDITION
{SS|STARTofframe|MISO|MOSI|MISOMOSI}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:CONDITION?

Related Commands
BUS:B<x>:SPI:IDLETime, BUS:B<x>:SPI:FRAMING

Arguments
SS specifies a search based on 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.

MISO specifies a search based on the Master-In Slave-Out condition.

MOSI specifies a search based on the Master-Out Slave-In condition.

MISOMOSI specifies a search based on the Master-In Slave-Out and Master-Out Slave-In conditions.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATa{:MISO|:IN}:VALue
Sets or returns the binary data string for an SPI trigger search if the search condition is MISO or MISOMOSI. SEARCH<x> is the search number and B<x> is the bus number.

Conditions
This command requires a DPO4EMBD application module.

Group
Search

Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATa{:MISO|:IN}:VALue <bin>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATa{:MISO|:IN}:VALue?

Arguments
<bin> is the data string in binary format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATa{:MOSI|:OUT}:VALue
Sets or returns the binary data string for an SPI trigger search if search the condition is MOSI, or MISOMOSI. SEARCH<x> is the search number and B<x> is the bus number.
Conditions
This command requires a DPO4EMBD application module.

Group
Search

Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATa{:MOSI|:OUT}:VALue <bin>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATa{:MOSI|:OUT}:VALue?

Arguments
<binc> is the data in binary format.

Sets or returns the length of the data string for an SPI trigger search if the search condition is MISO, MOSI, or MISOMOSI. SEARCH<x> is the search number and B<x> is the bus number.

Conditions
This command requires a DPO4EMBD application module.

Group
Search

Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATa:SIZe?

Arguments
<NR1> is the data string length in bytes.

Sets or returns the binary address string for the upper limit for inside-of-range and outside-of-range qualifiers for the USB search trigger. Use the command SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:ADDRess:VALue to set the lower limit.

Conditions
This command requires the DPO4USB application module.

Group
Search
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:ADDRess:HIVALue?

Related Commands
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:CONDition,

Arguments
<QString> within the range 0000000 to 1111111 (00 hex to 7F hex).

Examples
“0001000” sets the upper limit to binary 0001000 (08 hex).
“1111111”, which indicates that the upper limit is 1111111 (7F hex).


Sets or returns the binary address string to be used for the USB search trigger.
This command also sets or returns the binary address string for the lower limit for inside-of-range and outside-of-range qualifiers for the USB search trigger.

Conditions
This command requires the DPO4USB application module.

Group
Search

Syntax
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:ADDRess:VALue?

Related Commands
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:CONDition

Arguments
<QString> within the range 0000000 to 1111111 (00 hex to 7F hex).

Examples
sets the binary address to 0001000 (08 hex).
“1000000”, which indicates that the binary address is 1000000 (40 hex).
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:CONDition

Sets or returns the trigger condition for the USB search.

Conditions
This command requires the DPO4USB application module.

Group
Search

Syntax
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:CONDition
{SYNC|RESET|SUSPEND|RESUME|EOP|TOKENPacket|DATAPacket
|HANDSHAKEPacket|SPECIALPacket|ERROR}
SEARCH:SEARCH<x>:TRIGger:A: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
the trigger condition to be a token packet.

indicates that the trigger condition is a sync field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:DATa:HIVALue

Sets or returns the binary data string for the upper limit for inside-of-range
and outside-of-range qualifiers for the USB search trigger when trigger search
condition is DATAPacket. Use the command SEARCH:SEARCH<x>:TRIGger:
A:BUS:B<x>:USB:DATa:VALue to set the lower limit.
**SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:DATa:HIVALue**

Sets the upper limit to a specified value. The minimum and default value is 0, and the maximum is FFF (255 hex).

### Conditions
This command requires the DPO4USB application module.

### Group
Search

### Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:DATa:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:DATa:HIVALue?
```

### Arguments
<QString> within the range 00000000 to 11111111 (00 hex to FF hex).

### Examples
```
```
sets the upper limit to binary 00001000 (08 hex).
```
```
which indicates that the upper limit is 01111111 (7F hex).


Sets or returns the data offset in bytes to trigger the search on. The minimum and default values are 0 and the maximum is 1024.

### Conditions
This command requires the DPO4USB application module.

### Group
Search

### Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:DATa:OFFSet <NR1>
```

### Arguments
<NR1> is number in the range 0 to 1024.

### Examples
```
```
sets the data offset to 36.
```
```
indicates that the data offset is the default value, 0.

Sets or returns the number of contiguous data bytes to trigger the search on. The minimum and default values are 1 and the maximum value is 16.

**Conditions**
This command requires the DPO4USB application module.

**Group**
Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:USB:DATA:SIZE <NR1>
```

**Arguments**

<NR1> is number in the range 1 to 16.

**Examples**

```
```

sets the oscilloscope to trigger on four contiguous data bytes.

```
```

```
```

the oscilloscope will trigger on 6 contiguous data bytes.


Sets or returns the data type for the search trigger when the search trigger condition is set to DATAPacket.

**Conditions**
This command requires the DPO4USB application module.

**Group**
Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:USB:DATA:TYPE {ANY|DATA0|DATA1}
```

**Related Commands**

```
SEARCH:SEARCH<x>:TRIGGER:A: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.
Examples  
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:DATA:TYPE DATA0 sets the oscilloscope to trigger on a DATA0 data packet type.


Sets or returns the binary data string to be used for the USB search trigger when the search trigger condition is DATAPacket.

This command also sets or returns the binary data string for the lower limit for inside-of-range and outside-of-range qualifiers for USB search trigger when the trigger search condition is DATAPacket.

Conditions  
This command requires the DPO4USB application module.

Group  
Search

Syntax  

Related Commands  
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:USB:CONDITION

Arguments  
<QString> within the range 00000000 to 11111111 (00 hex to FF hex).

Examples  


Sets or returns the binary endpoint string to be used for the USB search trigger.

Conditions  
This command requires the DPO4USB application module.

Group  
Search
### SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:USB:ENDPoint:VALUE

**Syntax**

```
```

**Related Commands**

```
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:USB:CONDition
```

**Arguments**

`<QString>` within the range 0000 to 1111 (00 hex to 0F hex).

**Examples**

```
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:ENDPOINT:VALUE "1000" sets the binary address to 1000 (08 hex).

```

### SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:USB:ERRORTYPE

Sets or returns the error type used when the search trigger condition is set to ERROR.

**Conditions**

This command requires the DPO4USB application module.

**Group**

Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:USB:ERRORTYPE {PID|CRC5|CRC16|BITSTUFFing}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:USB:ERRORTYPE?
```

**Related Commands**

```
SEARCH:SEARCH<x>:TRIGGER:A: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**

```
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:CONDITION PID sets the error trigger condition to packet ID.
```
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:USB:HANDSHAKEType

Sets or returns the handshake type for the USB search trigger.

**Conditions**
This command requires the DPO4USB application module.

**Group**
Search

**Syntax**
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:USB:HANDSHAKEType
{ANY|NAK|ACK|STALL}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:USB:HANDSHAKEType?

**Related Commands**
SEARCH:SEARCH<x>:TRIGGER:A: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.

**Examples**


SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:USB:QUALifier

Sets or returns the USB search trigger qualifier for address, endpoint and data.

**Conditions**
This command requires the DPO4USB application module.

**Group**
Search
Commands Listed in Alphabetical Order

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:QUALifier
{LESSThan|MOREThan|EQUAL|UNEQual|LESSEQual
|MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:QUALifier?

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:CONDition

Arguments

LESSThan triggers the search on an value that is less than a set value.
MOREThan triggers the search on an value that is greater than a set value.
EQUAL triggers the search on an value that is to a set value.
UNEQual triggers the search on an value that is not equal to a set value.
LESSEQual triggers the search on an value that is less than or equal to a set value.
MOREEQual triggers the search on an value that is that is more than or equal to a set value.
INrange triggers the search on an value that is within a range set by two values.
OUTrange triggers the search on an value that is outside of a range set by two values.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:QUALIFIER LESSTHAN sets the oscilloscope to trigger when an address, data, or endpoint value is less than a set value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:SOFFRAME NUMber

Sets or returns the binary data string to be used for start of frame number, when the search trigger condition is Token Packet and the token type is Start of Frame.

Conditions

This command requires the DPO4USB application module.

Group

Search

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:SOFFRAME NUMber
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:USB:SOFFRAME NUMber?
Related Commands

SEARCH:SEARCH\(<x>\):TRIGGER:A:BUS:B<\>USB:CONDITION

Arguments

\(<\text{QString}\rangle\) within the range 000 0000 0000 to 111 1111 1111 (000 hex to 7FF hex).

Examples

“00000001000” sets the start of frame number to 00000001000 (008 hex).

“00000001001”, which indicates that the start of frame number is 00000001001 (009 hex).

SEARCH:SEARCH\(<x>\):TRIGGER:A:BUS:B<\>USB:SPECIALTYPE

Sets or returns the packet ID (PID) for the special packet.

Conditions

This command requires the DPO4USB application module.

Group

Search

Syntax

SEARCH:SEARCH\(<x>\):TRIGGER:A:BUS:B<\>USB:SPECIALTYPE
\{\text{ANY}|\text{PREamble}|\text{RESERVED}\}

SEARCH:SEARCH\(<x>\):TRIGGER:A:BUS:B<\>USB:SPECIALTYPE?

Arguments

\text{ANY} indicates any type of special packet.

\text{PREamble} indicates a preamble special packet.

\text{RESERVED} indicates a reserved special packet.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS1:USB:SPECIALTYPE PREAMBLE sets the special packet type to preamble.

:SEARCH:SEARCH1:TRIGGER:A:BUS1:USB:SPECIALTYPE PREAMBLE, indicating that the special type is set to preamble.

SEARCH:SEARCH\(<x>\):TRIGGER:A:BUS:B<\>USB:TOKENTYPE

Sets or returns the token type for the USB search trigger.
**Conditions**  This command requires the DPO4USB application module.

**Group**  Search

**Syntax**  
```
SEARCH:SEARCH<x>::TRIGGER:A:BUS:B<x>:USB:TOKENType
{ANY|SOF|OUT|IN|SETUP}
SEARCH:SEARCH<x>::TRIGGER:A:BUS:B<x>:USB:TOKENType?
```

**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**  
```
SEARCH:SEARCH1::TRIGGER:A:BUS:B1:USB:TOKENType SETUP sets the
token type to SETUP.
:SEARCH:SEARCH1::TRIGGER:A:BUS:B1:USB:TOKENType SOF if the token
type is SOF.
```

**SEARCH:SEARCH<x>::TRIGGER:A:BUS:SOURce**

Sets or returns a bus serial search.  `<x>` is the search number.

**Conditions**  This command requires a DPO4AUTO or DPO4EMBD application module.

**Group**  Search

**Syntax**  
```
SEARCH:SEARCH<x>::TRIGGER:A:BUS:SOURce {B1|B2|B3|B4}
SEARCH:SEARCH<x>::TRIGGER:A:BUS:SOURce?
```

**Arguments**  
- **B1** specifies the Bus 1 source.
- **B2** specifies the Bus 2 source.
- **B3** specifies the Bus 3 source.
- **B4** specifies the Bus 4 source.
**SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe**

Sets or returns the slope for an edge trigger search to determine where to place a mark. `<x>` is the search number.

**Group**  
Search

**Syntax**

SEND:SEARCH<x>:TRIGger:A:EDGE:SLOpe {RISe|FALL}
SEND:SEARCH<x>:TRIGger:A:EDGE:SLOpe?

**Arguments**

RISe specifies a rising edge.
FALL specifies a falling edge.

**SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce**

Sets or returns the source waveform for an edge trigger search to determine where to place a mark. `<x>` is the search number.

**Group**  
Search

**Syntax**

SEND:SEARCH<x>:TRIGger:A:EDGE:SOUrce {CH1|CH2|CH3|CH4|MATH}
SEND:SEARCH<x>:TRIGger:A:EDGE:SOUrce?

**Arguments**

CH<x> specifies one input channel as the edge source, where `<x>` is the channel number.

MATH specifies the math waveform as the search source.

**SEARCH:SEARCH<x>:TRIGger:A:LEVel**

Sets or returns the level for an edge trigger search to determine where to place a mark. `<x>` is the search number.

**Group**  
Search

**Syntax**

SEND:SEARCH<x>:TRIGger:A:LEVel {<NR3>|TTL}
SEND:SEARCH<x>:TRIGger:A:LEVel?
Arguments  

<NR3> specifies the trigger level, in volts.  
TTL specifies a preset TTL high level of 1.4V.

**SEARCH:SEARCH<x>:TRIGger:A:LEVel:CH<x>**

Sets or returns the level for an edge trigger search to determine where to place a mark. SEARCH<x> is the search number and CH<x> is the channel number. Each channel can have an independent level.

**Group**  
Search

**Syntax**  
SEARCH:SEARCH<x>:TRIGger:A:LEVel:CH<x> {<NR3>|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:LEVel:CH<x>?  

**Arguments**  

<NR3> specifies the trigger level in volts.  
TTL specifies a preset TTL high level of 1.4 V.

**SEARCH:SEARCH<x>:TRIGger:A:LEVel:MATH[1]**

Sets or returns the math waveform level for an edge trigger search to determine where to place a mark. <x> is the search number. The value of MATH is 1 for all oscilloscopes.

**Group**  
Search

**Syntax**  
SEARCH:SEARCH<x>:TRIGger:A:LEVel:MATH {TTL}  
SEARCH:SEARCH<x>:TRIGger:A:LEVel:MATH?  

**Arguments**  

TTL specifies a preset TTL high level of 1.4 V.

**SEARCH:SEARCH<x>:TRIGger:A:LEVel:REF<x>**

Sets or returns the specified reference waveform level for an edge trigger search to determine where to place a mark. SEARCH<x> is the search number and REF<x> is the reference channel number.

**Group**  
Search
SEARCH:SEARCH<x>:TRIGger:A:LEVel:REF<x> {TTL}
SEARCH:SEARCH<x>:TRIGger:A:LEVel:REF<x>?

Arguments
TTL specifies a preset TTL high level of 1.4 V.

**SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCtion**

Sets or returns the logic operator for a logic trigger search to determine where to place a mark. <x> is the search number.

**Group**
Search

**Syntax**
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.

**SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CH<x>**

Sets or returns the Boolean logic criteria for a logic trigger search to determine where to place a mark. SEARCH<x> is the search number and CH<x> is the channel number.

**Group**
Search

**Syntax**
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CH<x> {HIGH|LOW|X}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CH<x>?

**Arguments**
HIGH specifies the logic high.
LOW specifies the logic low.
X specifies a "don’t care" state.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CLOCk:EDGE

Sets or returns whether the clock edge is a rising or falling for a logic search to determine where to place a mark. <x> is the search number.

**Group**  Search

**Syntax**  SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CLOCk:EDGE
{FALL|RISe}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CLOCk:EDGE?

**Arguments**  RISe specifies a rising edge.
FALL specifies a falling edge.


Sets or returns the clock source definition for a logic trigger search. <x> is the search number. If a clock source is defined, then the logic search is determined by the state of the other inputs at the clock transition. If no clock source is defined, then the logic search is determined only by the state of the inputs.

**Group**  Search

**Syntax**  SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CLOCk:SOUrce
{CH1|CH2|CH3|CH4|MATH|REF|NONe}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CLOCk:SOUrce?

**Arguments**  CH<x> specifies a channel input as the clock source, where <x> = 1, 2, 3, or 4.
MATH specifies the math waveform as the clock source.
REF specifies the reference waveform as the clock source.
NONe specifies no clock source.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:D<x>

Sets or returns the waveform logic criteria for a trigger search. SEARCH<x> is the search number and D<x> is the digital channel number.

**Group**  Search
| Syntax | SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:INPUT:D<x> {HIGH|LOW|X}  
SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:INPUT:D<x>? |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Arguments | HIGH specifies a high logic level.  
LOW specifies a low logic level.  
X specifies a “don’t care” condition. |

**SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:INPUT:MATH**

Sets or returns the Boolean logic criteria for a logic trigger search to determine where to place a mark. <x> is the search number.

<table>
<thead>
<tr>
<th>Group</th>
<th>Search</th>
</tr>
</thead>
</table>
| Syntax | SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:INPUT:MATH {HIGH|LOW|X}  
SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:INPUT:MATH? |
| Arguments | HIGH specifies a high logic level.  
LOW specifies a low logic level.  
X specifies a “don’t care” condition. |

**SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:INPUT:REF<x>**

Sets or returns the Boolean logic criteria for a logic trigger search to determine where to place a mark. SEARCH<x> is the search number and REF<x> is the reference channel number.

<table>
<thead>
<tr>
<th>Group</th>
<th>Search</th>
</tr>
</thead>
</table>
| Syntax | SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:INPUT:REF<x> {HIGH|LOW|X}  
SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:INPUT:REF<x>? |
| Arguments | HIGH specifies a high logic level.  
LOW specifies a low logic level.  
X specifies a “don’t care” condition. |
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAIttern:INPut:CH<x>

Sets or returns the logic criteria for a logic pattern trigger search to determine where to place a mark. SEARCH<x> is the search number and CH<x> is the channel number.

**Group** Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAIttern:INPut:CH<x> {HIGH|LOW|X}

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAIttern:INPut:CH<x>?

**Arguments**

HIGH specifies a high logic level.

LOW specifies a low logic level.

X specifies a “don’t care” condition.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAIttern:INPut:D<x>

Sets or returns the logic criteria for a logic pattern search to determine where to place a mark. SEARCH<x> is the search number and D<x> is the digital channel number.

**Group** Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAIttern:INPut:D<x> {HIGH|LOW|X}

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAIttern:INPut:D<x>?

**Arguments**

HIGH specifies a high logic level.

LOW specifies a low logic level.

X specifies a “don’t care” condition.


Sets or returns the Boolean logic criteria for a logic pattern trigger search to determine where to place a mark. SEARCH<x> is the search number.

**Group** Search
{HIGH|LOW|X}
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:PArttern:INPut:MATH?

Arguments
HIGH specifies a high logic level.
LOW specifies a low logic level.
X specifies a “don’t care” condition.

SEARCH:SEARCH<x>:TRIGger:A:LOGIC:PArttern:INPut:REF<x>

Sets or returns the Boolean logic criteria for a pattern trigger search to determine where to place a mark. SEARCH<x> is the search number and REF<x> is the reference channel number.

Group
Search

Syntax
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:PArttern:INPut:REF<x> {HIGH|LOW|X}
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:PArttern:INPut:REF<x>?

Arguments
HIGH specifies a high logic level.
LOW specifies a low logic level.
X specifies a “don’t care” condition.

SEARCH:SEARCH<x>:TRIGger:A:LOGIC:PArttern:WHen

Sets or returns the condition for generating a logic pattern trigger search to determine where to place a mark. <x> is the search number.

Group
Search

Syntax
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:PArttern:WHen
{TRUE|FALSE|LESSThan|Than|EQUal|UNEQual}
SEARCH:SEARCH<x>:TRIGger:A:LOGIC:PArttern:WHen?

Arguments
TRUE places a mark when the pattern becomes true.
FALSE places a mark when the pattern becomes false.
**LESSThan** places a mark if the specific pattern is true less than the time set by the `SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAttern:WHEn:LESSLimit` command.

**Than** places a mark if the specific pattern is true longer than the specified time set by the `SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAttern:WHEn:MORELimit` command.

**EQUa1** places a mark if the specific pattern is true longer than the time set by the `SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAttern:WHEn:LESSLimit` command, but less than the specified time set by the `SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAttern:WHEn:MORELimit` command.

**UNEQual** places a mark if the specific pattern is true less than the time set by the `SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAttern:WHEn:LESSLimit` command, or longer than the specified time set by the `SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAttern:WHEn:MORELimit` command.

**SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAttern:WHEn:LESSLimit**

Sets or returns the maximum time that the selected pattern may be true and still generate an A logic pattern search to place a mark. `<x>` is the search number.

**Group** Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAttern:WHEn:LESSLimit
<NR3>
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAttern:WHEn:LESSLimit?
```

**Arguments**

`<NR3>` specifies the maximum amount of time to hold the pattern true.

**SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAttern:WHEn:MORELimit**

Sets or returns the minimum time that the selected pattern may be true and still generate an A logic pattern search to place a mark. `<x>` is the search number.

**Group** Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAttern:WHEn:MORELimit
<NR3>
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PAttern:WHEn:MORELimit?
```
Arguments  <NR3> specifies the minimum amount of time to hold the pattern true.

**SEARCH:SEARCH<x>:TRIGger:A:LOGIc:THReshold:CH<x>**

Sets or returns the channel threshold level for a logic trigger search to determine where to place a mark. SEARCH<x> is the search number and CH<x> is the channel number.

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:THReshold:CH<x> {<NR3>|TTL}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:THReshold:CH<x>?  

**Arguments**  

<NR3> specifies the trigger level, in volts.  
TTL specifies a preset TTL high level of 1.4 V.

**SEARCH:SEARCH<x>:TRIGger:A:LOGIc:THReshold:MATH**

Sets or returns the math waveform threshold level for a logic trigger search to determine where to place a mark. <x> is the search number.

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:THReshold:MATH {TTL}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:THReshold:MATH?  

**Arguments**  

TTL specifies a preset TTL high level of 1.4 V.

**SEARCH:SEARCH<x>:TRIGger:A:LOGIc:THReshold:REF<x>**

Sets or returns the reference waveform threshold level for a logic trigger search to determine where to place a mark. SEARCH<x> is the search number and REF<x> is the reference channel number.

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:THReshold:REF<x> {TTL}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:THReshold:REF<x>?
Arguments  TTL specifies a preset TTL high level of 1.4 V.

**SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:CH<x>**

Sets or returns the channel waveform lower threshold to determine where to place a mark. This setting is applied to all channel searches that use a lower threshold. **SEARCH<x>** is the search number and **CH<x>** is the channel number.

**Group**  Search

**Syntax**  SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:CH<x> {TTL}
SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:CH<x>?  

**Arguments**  TTL specifies a preset TTL high level of 1.4 V.

**SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:MATH**

Sets or returns the math waveform lower threshold to determine where to place a mark. This setting is applied to all math searches that use a lower threshold. **<x>** is the search number.

**Group**  Search

**Syntax**  SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:MATH {TTL}
SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:MATH?  

**Arguments**  TTL specifies a preset TTL high level of 1.4 V.

**SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:REF<x>**

Sets or returns the reference waveform lower threshold to determine where to place a mark. This setting is applied to all reference searches that use a lower threshold. **SEARCH<x>** is the search number and **REF<x>** is the reference channel number.

**Group**  Search
Syntax

```
SEARCH:SEARCH<x>:TRIGGER:A:LOWerthreshold:REF<x> {TTL}
SEARCH:SEARCH<x>:TRIGGER:A:LOWerthreshold:REF<x>?
```

Arguments

TTL specifies a preset TTL high level of 1.4 V.

**SEARCH:SEARCH<x>:TRIGGER:A:PULSEwidth:POLarity**

Sets or returns the polarity for a pulse trigger search to determine where to place a mark. <x> is the search number.

**Group**
Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGGER:A:PULSEwidth:POLarity {NEGative|POSitive}
SEARCH:SEARCH<x>:TRIGGER:A:PULSEwidth:POLarity?
```

Arguments

POSITIVE places a mark only when the polarity of the pulse is positive.

NEGATIVE places a mark only when the polarity of the pulse is negative.

**SEARCH:SEARCH<x>:TRIGGER:A:PULSEwidth:SOURce**

Sets or returns the source waveform for a pulse trigger search to determine where to place a mark. <x> is the search number.

**Group**
Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGGER:A:PULSEwidth:SOURce {CH1|CH2|CH3|CH4|MATH|REF}
SEARCH:SEARCH<x>:TRIGGER:A:PULSEwidth:SOURce?
```

Arguments

CH<x> specifies one input channel as the edge source, where <x> = 1, 2, 3 or 4.

MATH specifies the math waveform as the search source.

REF specifies the reference waveform as the search source.

**SEARCH:SEARCH<x>:TRIGGER:A:PULSEwidth:WHEn**

Sets or returns the condition for generating a pulse width search to determine where to place a mark. <x> is the search number.
Group  
Search

Syntax  
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WHEn
{LESSThan|than|EQual|UNEQual}
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WHEn?

Arguments  
LESSThan places a mark if the pulse width is less than the time set by the

Than places a mark if the pulse width is true longer than the specified time set by
the SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WIDth command.

EQUal places a mark if the pulse width is equal to the time set by the
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WIDth command within a
tolerance of ±5%.

UNEQual places a mark if the pulse width is unequal to the time the time set by
the SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WIDth command within a
tolerance of ±5%.

SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WIDth

Sets or returns the pulse width setting for a pulse width trigger search to determine
where to place a mark. <x> is the search number.

Group  
Search

Syntax  
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WIDth <NR3>
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WIDth?

Arguments  
<NR3> is the pulse width.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity

Sets or returns the polarity setting for a runt trigger search to determine where to
place a mark. <x> is the search number.

Group  
Search

Syntax  
SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity
{EITher|NEGative|POSitive}
SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity?
Arguments

**POSitive** places a mark when the rising edge crosses the low threshold and the falling edge re-crosses the low threshold without either edge ever crossing the high threshold.

**NEGative** places a mark when the falling edge crosses the high threshold and the rising edge re-crosses the high threshold without either edge ever crossing the low threshold.

**EITHER** places a mark on a runt of either polarity.

**SEARCH:** SEARCH<x>:TRIGger:A:RUNT:SOURce

Sets or returns the source setting for a runt trigger search to determine where to place a mark. `<x>` is the search number.

**Group** Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOURce

{CH1|CH2|CH3|CH4|MATH|REF}

SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOURce?

**Arguments**

CH1–CH4 specifies an input channel as the edge source.

MATH specifies the math waveform as the search source.

REF specifies the reference waveform as the search source.

**SEARCH:** SEARCH<x>:TRIGger:A:RUNT:WHEn

Sets or returns the condition setting for a runt trigger search to determine where to place a mark. `<x>` is the search number.

**Group** Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn

{LESSthan|than|EQual|UNEQual|OCCURS}

SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn?

**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 SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth command.
than argument sets the oscilloscope to trigger if the a runt pulse is detected with width than the time set by the SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth command.

EQUAL argument sets the oscilloscope to trigger 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 trigger 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.

**SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth**

Sets or returns the width setting for a runt trigger search to determine where to place a mark. <x> is the search number.

**Group** Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth <NR3>

SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth?

**Arguments**

<NR3> specifies the minimum width, in seconds.

**SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:EDGE**

Sets or returns the clock slope setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number.

**Group** Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:EDGE {FALL|RISe}

SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:EDGE?

**Arguments**

FALL specifies polarity as the clock falling edge.

RISe specifies polarity as the clock rising edge.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:SOUrce

Sets or returns the clock source setting for an setup/hold trigger search to
determine where to place a mark. <x> is the search number.

Syntax
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:SOUrce
{CH1|CH2|CH3|CH4|MATH|REF}
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:SOUrce?

Arguments
CH1–CH4 specifies an input channel as the edge source.
MATH specifies the math waveform as the search source.
REF specifies the reference waveform as the search source.

Related Commands
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATa:SOUrce

SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:THReshold

Sets or returns the clock threshold setting for an setup/hold trigger search to
determine where to place a mark. <x> is the search number.

Syntax
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:THReshold
{<NR3>|TTL|ECL}
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:THReshold?

Arguments
TTL specifies a preset TTL high level of 1.4 V.
ECL specifies a preset ECL high level of -1.3V.
<NR3> is the clock level, in volts.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATa:SOUrce

Sets or returns the data source setting for an setup/hold trigger search to determine
where to place a mark. <x> is the search number. You cannot specify the same
source for both clock and data.
Commands Listed in Alphabetical Order

**Group** Search

**Syntax**

**DPO Models:**

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATa:SOURce
{CH1|CH2|CH3|CH4|MATH|REF}
```

**MSO Models:**

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATa:SOURce <wfm>[,<wfm>] [<wfm>] [<wfm>] [<wfm>] [<wfm>] [<wfm>] [<wfm>] [<wfm>] [<wfm>] [<wfm>] [<wfm>] [<wfm>]
```

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATa:SOURce?
```

**Related Commands**

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:SOURce
```

**Arguments**

**DPO Models:**

CH1–CH4 specifies an input channel as the search source.

MATH specifies the math waveform as the search source.

REF specifies the reference waveform as the search source.

**MSO Models:**

<wfm> can be any combination of the channel, math, reference, and digital waveforms.

**SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATa:THReshold**

Sets or returns the data threshold setting for an setup/hold trigger search to determine where to place a mark. <x> is the search number.

**Group** Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATa:THReshold
{<NR3>|TTL}
```

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATa:THReshold?
```

**Arguments**

TTL specifies a preset TTL high level of 1.4 V.

<NR3> is the clock level, in volts.
**SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime**

Sets or returns the hold time setting for an setup/hold trigger search to determine where to place a mark. <x> is the search number.

**Group** Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime <NR3>
SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime?

**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.

**SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime**

Sets or returns the setup time setting for an setup/hold trigger search to determine where to place a mark. <x> is the search number.

**Group** Search

**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.

**SEARCH:SEARCH<x>:TRIGger:A:SETHold:THReshold:CH<x>**

Sets or returns the trigger search setup and hold lower threshold for the selected channel. This helps determine where to place search marks. Search<x> is the search number, which is always 1. CH<x> is the channel number.

**Conditions** MSO oscilloscopes only.

**Group** Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:SETHold:THReshold:CH<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:SETHold:THReshold:CH<x>?
Arguments

<NR3> is the lower threshold in volts.

**SEARCH:SEARCH<x>:TRIGger:A:SETHold:THReshold{:MATH|:MATH1}**

Sets or returns the trigger search setup and hold threshold for the math waveform. This helps to determine where to place search marks. Search<x> is the search number, which is always 1.

**Conditions**

MSO oscilloscopes only.

**Group**

Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:SETHold:THReshold{:MATH|:MATH1}

<NR3>

SEARCH:SEARCH<x>:TRIGger:A:SETHold:THReshold{:MATH|:MATH1}?/n

**Arguments**

<NR3> is the lower threshold in volts.

**SEARCH:SEARCH<x>:TRIGger:A:SETHold:THReshold:REF<x>**

Sets or returns the trigger search setup and hold threshold for the selected reference waveform. This helps to determine where to place search marks. Search<x> is the search number, which is always 1. REF<x> is the reference waveform number.

**Conditions**

MSO oscilloscopes only.

**Group**

Search

**Syntax**

SEARCH:SEARCH<x>:TRIGger:A:SETHold:THReshold:REF<x> <NR3>

SEARCH:SEARCH<x>:TRIGger:A:SETHold:THReshold:REF<x>?/n

**Arguments**

<NR3> is the lower threshold in volts.

**SEARCH:SEARCH<x>:TRIGger:A{:TRANSition|:RISEFall}:DELTatime**

Sets or returns the transition time setting for an transition trigger search to determine where to place a mark.<x> is the search number.
Group  
Search

Syntax  
SEARCH:SEARCH<x>:TRIGger:A{TRANSition|RISEFall}:DELTatime
<NR3>  
SEARCH:SEARCH<x>:TRIGger:A{TRANSition|RISEFall}:DELTatime?

Arguments  
<NR3> specifies the transition time, in seconds.

**SEARCH:SEARCH<x>:TRIGger:A{TRANSition|RISEFall}:POLarity**

Sets or returns the polarity setting for a transition trigger search to determine where to place a mark. <x> is the search number.

Group  
Search

Syntax  
SEARCH:SEARCH<x>:TRIGger:A{TRANSition|RISEFall}:POLarity
{EITHER|NEGative|POSitive}
SEARCH:SEARCH<x>:TRIGger:A{TRANSition|RISEFall}:POLarity?

Arguments  
POSitive specifies that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for transition triggering to occur.

NEGative specifies that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.

EITHER specifies either positive or negative polarity.

**SEARCH:SEARCH<x>:TRIGger:A{TRANSition|RISEFall}:SOUrce**

Sets or returns the source setting for a transition trigger search to determine where to place a mark. <x> is the search number.

Group  
Search

Syntax  
SEARCH:SEARCH<x>:TRIGger:A{TRANSition|RISEFall}:SOUrce
{CH1|CH2|CH3|CH4|MATH}
SEARCH:SEARCH<x>:TRIGger:A{TRANSition|RISEFall}:SOUrce?

Arguments  
CH1–CH4 specifies one input channel as the edge source.

MATH specifies the math waveform as the search source.
SEARCH:SEARCH<x>:TRIGger:A{:TRANsition|:RISEFall}:WHEn

Sets or returns the condition setting for an transition trigger search to determine where to place a mark. <x> is the search number.

Group Search

Syntax

SEARCH:SEARCH<x>:TRIGger:A{:TRANsition|:RISEFall}:WHEn {SLOWer|FASTER|EQUAL|UNEQual}
SEARCH:SEARCH<x>:TRIGger:A{:TRANSition|:RISEFall}:WHEn?

Arguments

FASTER sets the trigger to occur when the transitioning signal is faster than the set volts/second rate.

SLOWer 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 within a ±5% tolerance.

UNEQual sets the trigger to occur when the transitioning signal is not equal to the set volts/second rate ±5%.

SEARCH:SEARCH<x>:TRIGger:A:TYPe

Sets or returns the trigger type setting for a search to determine where to place a mark. <x> is the search number.

Group Search

Syntax

SEARCH:SEARCH<x>:TRIGger:A:TYPe {EDGEdge|SETHold|PULSEWidth|RUNt|TRANSition|LOGIc|BUS (with the appropriate application module installed)}
SEARCH:SEARCH<x>:TRIGger:A:TYPe?

Arguments

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:LOWerthreshold:CH<x> and SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:CH<x> commands.

PULSEWidth triggers when a pulse is found that has the specified polarity and is either inside or outside the limits as specified by SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATHterm:WHEn:LESSLimit and SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATHterm:WHEn:MORELimit.
The polarity is selected using the `SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity` command.

`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.

**SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:CH<x>**

Sets or returns the channel waveform upper threshold to determine where to place a mark. This setting is applied to all channel searches that uses an upper threshold. `SEARCH<x>` is the search number and `CH<x>` is the channel number.

**Group** Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:CH<x> {TTL}
SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:CH<x>?
```

**Arguments**

`TTL` specifies a preset TTL high level of 1.4 V.

**SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:MATH**

Sets or returns the math waveform upper threshold to determine where to place a mark. This setting is applied to all math waveform searches that uses an upper threshold. `<x>` is the search number.

**Group** Search

**Syntax**

```
SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:MATH {TTL}
SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:MATH?
```

**Arguments**

`TTL` specifies a preset TTL high level of 1.4 V.

**SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:REF<x>**

Sets or returns the reference waveform upper threshold to determine where to place a mark. This setting is applied to all reference waveform searches that uses an upper threshold. `SEARCH<x>` is the search number and `REF<x>` is the reference channel number.
Commands Listed in Alphabetical Order

**Group**  
Search

**Syntax**  
SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:REF<x> {TTL}  
SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:REF<x>?  

**Arguments**  
TTL specifies a preset TTL high level of 1.4 V.

**SESelect**

Sets or returns the selected waveform display (controlled by the front-panel) on or off.

**Group**  
Vertical

**Syntax**  
SELECT {ON|OFF}

**Arguments**  
ON turns the selected waveform display on.  
OFF turns the selected waveform display off.

**Examples**  
SELECT might return the following  
:SELECT:BUS1 0;BUS2 0;CH1 1;CH2 0;CH3 0;CH4 0;MATH 0;REF1 0;REF2 0;REF3 0;REF4 0;CONTROL CH1

**SESelect:BUS<x>**

This command turns on and off the display of the waveform for <x>, where x is the bus number. The query returns whether the channel is on or off but does not indicate whether it is the selected waveform.

**Group**  
Vertical

**Syntax**  
SELECT:BUS<x> {<NR1>|OFF|ON}  
SELECT:BUS<x>?
**SELect:CH<x>**

Turns the display of the channel `<x>` waveform on or off, where `<x>` is the channel number. This command also resets the acquisition. The query returns whether the channel is on or off but does not indicate whether it is the selected waveform.

**Group**  
Vertical

**Syntax**  
```
SELect:CH<x> {ON|OFF|<NR1>}
SELect:CH<x>?
```

**Arguments**  
- **ON** turns on the display of the specified waveform. This waveform also becomes the selected waveform.
- **OFF** turns off the display of the specified waveform.
- `<NR1> = 0` turns off the display of the specified waveform; any other value turns on the display of the specified waveform.

**Examples**  
```
SELECT:CH2 ON turns the channel 2 waveform display on, and selects channel 2.
SELECT:CH1? might return :SELECT:CH1 1 indicating that channel 1 is being displayed.
```

**SELect:CONTROL**

Sets or returns the waveform that is the recipient of future channel-related commands, for example, the cursor commands. The command form also performs the equivalent of a **SELect:CH<x>** ON command, as well as the Math, Reference, and Bus variations of that command.

**Group**  
Vertical

**Syntax**  
```
SELect:CONTROL {CH<x>|MATH|BUS<x>}
SELect:CONTROL?
```

**Arguments**  
- **CH<x>** specifies a channel waveform as the waveform affected by the front-panel controls. `<x>` is the channel number.
- **MATH** specifies the math waveform as the waveform that is affected by the front-panel controls.
BUS<x> specifies a bus waveform as the waveform affected by the front-panel controls.  <x> specifies the bus number.

**Returns**

NONE if all the channels are turned off.  NONE is ignored on input.

**Examples**

SELECT:CONTROL CH2 resets acquisition displays on channel 2, and causes the selected waveform to be the implied object of waveform commands.

SELECT:CONTROL? might return :SELECT:CONTROL MATH indicating that math is the implied object of waveform commands.

**SELect:D<x>**

Turns on the display of the digital channel <x> and resets the acquisition.  <x> is the channel number.  The query returns whether the channel is on or off but does not indicate whether it is the selected waveform.

**Group**

Vertical

**Syntax**

SELect:D<x> {<NR1>|OFF|ON}

SELect:D<x>?

**Arguments**

**ON** turns on the display of the specified waveform.  This waveform also becomes the selected waveform.

**OFF** turns off the display of the specified waveform.

**<NR1> = 0** turns off the display of the specified waveform; any other value turns on the display of the specified waveform.

**Examples**

SELECT:D2  ON turns the digital channel 2 waveform display on, and selects digital channel 2.

SELECT:D2? might return :SELECT:D2 1 indicating that digital channel 2 is being displayed.

**SELect:MAT[1]**

Turns on and off the display of the math waveform.  The query returns whether the math waveform is on or off but does not indicate whether it is the selected waveform.
Commands Listed in Alphabetical Order

**Group** Vertical

**Syntax**

```
SELect:MATH[1] {ON|OFF|<NR1>}
SELect:MATH[1]?
```

**Arguments**

ON turns on the display of the specified waveform. This waveform also becomes the selected waveform.

OFF turns off the display of the specified waveform.

<NR1> = 0 turns off the display of the specified waveform; any other value turns on the display of the specified waveform.

**Examples**

```
SELECT:MATH ON turns the math waveform display on, and selects it.
SELECT:MATH? might return :SELECT:MATH 1 indicating that the math waveform is being displayed.
```

**SELect:REF<x>**

Turns on and off the display of the reference waveform <x>. The <x> variable represents the reference channel number. The query returns whether the channel is on or off.

**Group** Vertical

**Syntax**

```
SELect:REF<x> {ON|OFF|<NR1>}
SELect:REF<x>?
```

**Arguments**

ON turns on the display of the specified waveform. This waveform also becomes the selected waveform.

OFF turns off the display of the specified waveform.

<NR1> = 0 turns off the display of the specified waveform; any other value turns on the display of the specified waveform.

**Examples**

```
SELECT:REF2 ON turns the channel 2 waveform display on, and selects reference waveform 2.
SELECT:REF2? might return :SELECT:REF2 1 indicating that reference waveform 2 is being displayed.
```
SET? (Query Only)

Returns the commands that list the oscilloscope settings except for configuration information for the calibration values, the WFMInpre? query, and the WFMOutpre? query. This query allows you to record or "learn" the current oscilloscope settings. You can use these commands to return the oscilloscope 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 oscilloscope 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? returns a long response, part of which could be as follows:

```
:SET:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV INFINITE;NUMAVG 16;SAMPLINGMODE RT;:HEADER 1;:LOCK NONE;:LANGUAGE ENGLISH;:VERBOSE 1;:ALIAS:STATE 0;;DISPLAY:COLOR:PALETTE NORMAL;;DISPLAY:STYLE:DOTSONLY 0;;DISPLAY:PERSISTENCE 0.0000;CLOCK 1;GRATICULE FULL;INTENSITY:WAVEFORM 30;GRATICULE 75;BACKLIGHT HIGH;;HARDCOPY:INKSAVER OFF;LAYOUT LANDSCAPE;PREVIEW 0;;SAVE:IMAGE:FILEFORMAT BMP;;SAVE:WAVEFORM:FILEFORMAT INTERNAL;;SAVE:ASSIGN:TYPE SETUP;;TRIGGER:A:MODE AUTO;TYPE EDGE;LEVEL 20.0000E-3;LEVEL:CH1 20.0000E-3;CH2 0.0000;CH3 0.0000;CH4 0.0000;;TRIGGER:A:UPPERTHRESHOLD:CH1 1.4000;CH2 800.0000E-3;CH3 800.0000E-3;CH4 800.0000E-3;;TRIGGER:A:LOWERTHRESHOLD:CH1 20.0000E-3;CH2 0.0000;CH3 0.0000;CH4 0.0000;;TRIGGER:A:HOLDOFF:TIME 20.0000E-9;;TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;;TRIGGER:A:LOGIC:CLASS SET HOLD;FUNCTION AND;THRESHOLD:CH1 20.0000E-3;CH2 0.0000;CH3 0.0000;CH4 0.0000;;TRIGGER:A:LOGIC:INPUT:CH1 X;CH2 X;CH3 X;CH4 X;;CLOCK:SOURCE NONE;EDGE RISE;;TRIGGER:A:LOGIC:PATTERN:INPUT:CH1 X;CH2 X;CH3 X;CH4 X;;TRIGGER:A :LOGIC:PATTERN:WHEN TRUE;WHEN:LESSLIMIT 4.0000E-9;LIMIT 4.0000E-9;;TRIGGER:A :SETHOLD:CLOCK:SOURCE CH1;EDGE RISE;THRESHOLD 20.0000E-3;;TRIGGER:A:SETHOLD:DATA:SOURCE CH2;
```
SETUP<x>:DATE? (Query Only)

Returns the date when the oscilloscope setup was saved for the specified channel <x>.

Group   Save and Recall
Syntax   SETUP<x>:DATE?
Examples SETUP4:DATE? might return SETUP4:DATE: 04–18–06 which is the setup date for channel 4.

SETUP<x>:LABEL

Sets or returns the setup label for the specified channel <x>, where x can be 1 through 10.

Group   Save and Recall
Syntax   SETUP<x>:LABEL <Qstring>
Arguments <Qstring> is an alphanumeric string of characters, enclosed in quotes, that defines the label text for SETUP<x>. The length of the string is limited to 30 characters.
Examples SETUP:LABEL? might return SETUP1:LABEL: TEST 2 which is the label setup for channel 1.

SETUP<x>:TIME? (Query Only)

Returns the time when the oscilloscope setup was saved for the specified channel <x>.

Group   Save and Recall
Syntax   SETUP<x>:TIME?
Examples

SETUP2:TIME? might return "SETUP2:TIME: 15:24:07 which is the setup time for channel 2.

*SRE

The *SRE (Service Request Enable) command sets or returns the bits in the Service Request Enable Register. For information, refer to Registers.

Group

Status and Error

Syntax

*SRE <NR1>
*SRE?

Related Commands


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.

*STB? (Query Only)

*STB? (Read Status Byte) returns the contents of the Status Byte Register (SBR) using the Master Summary Status (MSS) bit. For information, refer to Registers.

Group

Status and Error

Syntax

*STB?

Related Commands


Returns

<NR1>
Examples

*STB? might return 96, showing that the SBR contains the binary value 01100000.

TEKSecure (No Query Form)

This command initializes both waveform and setup memories, overwriting any previously stored data. These are the \texttt{WFMINpre?}, \texttt{WFMOutpre?}, and \texttt{DATA} command values after the \texttt{TEKSecure} operation.

\begin{verbatim}
\end{verbatim}

\textbf{NOTE.} The \texttt{TEKSecure} command can take up to five minutes to complete. The oscilloscope is inoperable during this period.

Group

Miscellaneous

Syntax

\texttt{TEKSecure}

Examples

\texttt{TEKSECURE} initializes both waveform and setup memories.

This is a program example of how to generate an SRQ when \texttt{TEKSECURE} completes:

\begin{verbatim}
# Bit 0 of the DESE (Device Event Status Enable Register) # enables OPC to be reported to the SESR (Standard Event Status Register) DESE 255 # Bit 0 of the ESER (Event Status Enable Register) # enables OPC to be summarized in the ESB (Event Status # Bit) of the SBR (Status Byte Register) *ESE 255
\end{verbatim}
# Bit 5 of the SRE (Service Request Enable Register)
# enables
# the generation of SRQ when the ESB bit of the SBR
becomes # TRUE
*SRE 32
TEKSECURE;*OPC

When the TEKSECURE operation has completed, the OPC bit of the SESR will be TRUE and SRQ will have been generated.

**TIME**

Sets or returns the time that the oscilloscope displays.

**Group**  
Miscellaneous

**Syntax**  
TIME <QString>
TIME?

**Related Commands**  
DATE

**Arguments**  
<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 00 to 59, and ss refers to a two-digit second number from 00 to 59.

**Examples**  
TIME "14:00:00" sets the time to exactly 2:00 p.m.
TIME? might return :TIME "14:05:17" indicating the current time is set to 2:05 p.m. and 17 seconds.

**TOTaluptime? (Query Only)**

This command returns the total number of hours that the oscilloscope has been powered on since the nonvolatile memory was last programmed (usually since the initial manufacturing process).

**Group**  
Miscellaneous

**Syntax**  
TOTaluptime?
**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 (No Query Form)**

Forces a trigger event to occur.

**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.

**Examples**  
TRIGGER FORCe forces a trigger event to occur.

**TRIgger:A**

Sets the A trigger level automatically to 50% of the range of the minimum and maximum values of the trigger input signal. The query returns current A trigger parameters. The trigger level is the voltage threshold through which the trigger source signal must pass to generate a trigger event. This command works for the following cases: Edge Trigger (when source is Not Line), Logic Trigger (when Clock Source is not Off or Logic Pattern is Don’t Care), and Pulse Width Trigger.

**Group**  
Trigger
Syntax

TRIGGER:A SETLevel
TRIGGER:A?

Related Commands


Arguments

SETLevel sets the A trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.

Examples

TRIGGER:A SETLEVEL sets the A trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.

TRIGGER:A? might return a long response with A trigger parameters, some of which could be as follows:

```
:TRIGGER:A:MODE AUTO;TYPE EDGE;LEVEL 20.0000E-3;LEVEL:CH1 20.0000E-3;CH2 0.0000;CH3 0.0000;CH4 0.0000;:TRIGGER:A:UPPERTHRESHOLD:CH1 1.4000;CH2 800.0000E-3;CH3 8 00.0000E-3;CH4 800.0000E-3;:TRIGGER:A:LOWERTHRESHOLD:CH1 20.0000E-3;CH2 0.0000;CH3 0.0000;CH4 0.0000;:TRIGGER:A:HOLDOFF:TIME 20.0000E-9;:TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;:TRIGGER:A:LOGIC:CLASS SETHOLD;FUNCTION AND;THRESHOLD: CH1 20.0000E-3;CH2 0.0000;CH3 0.0000;CH4 0.0000;:TRIGGER:A:LOGIC:INPUT:CH1 X;CH2 X;CH3 X;CH4 X;CLOCK:SOURCE NONE;EDGE RISE;:TRIGGER:A:LOGIC:PATTERN:INPUT:CH1 X; CH2 X;CH3 X;CH4 X;:TRIGGER:A:LOGIC:PATTERN:WHEN TRUE;WHEN:LESSLIMIT 4.0000E-9;MO RELIMIT 4.0000E-9;:TRIGGER:A:SETHOLD:CLOCK:SOURCE CH1;EDGE RISE;THRESHOLD 20.0000E-3;:TRIGGER:A:SETHOLD:DATA:SOURCE CH2;THRESHOLD 0.0000;:TRIGGER:A:SETHOLD:HOLD TIME 4.0000E-9;SETTIME 4.0000E-9;:TRIGGER:A:PULSE:CLASS TRANSITION;:TRIGGER:A:PULSEWIDTH:SOURCE CH1;POLARITY POSITIVE;WHEN LESS THAN;WIDTH 4.0000E-9;:TRIGGER:A:RUNT:SOURCE CH1;POLARITY POSITIVE;WHEN OCCURS;WIDTH 4.0000E-9;:TRIGGER:A:TRANSITION:SOURCE CH1;POLARITY POSITIVE;WHEN SLOWER;DELTA TIME 4.0000E-9;:TRIGGER:A:VIDEO:CLASS NTSC;SYNC ALL LINES;HOLDOFF:FIELD 0.0000;:TRIGGER:A:VIDEO:CUSTOM:FORMAT PROGRESSIVE;SCAN RATE15K;:TRIGGER:A:VIDEO:LINE 1;H DTV:FORMAT HD1080I60;:TRIGGER:A:BUS:SOURCE B1;B1:I2C:CONDITION START;DATA:VALUE "XXXXXXXX";SIZE 1;START 0.0000;DIRECTION NOCARE;:TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE ADDR7;TYPE USER;VALUE "XXXXXXXX";:TRIGGER:A:BUS:B1:SPI:CONDITION MOSI;DATA:OUT :VALUE "XXXXXXXX";:TRIGGER:A:BUS:B1:SPI:DATA:IN:VALUE "XXXXXXXX";:TRIGGER:A:BUS: B1:SPI:DATA:SIZE 1;START 0.0000;
```
**TRIGger:A:BUS**

Sets or returns the trigger type: I2C, CAN, SPI, RS-232, or Parallel. There are up to four serial buses, B1–B4, depending on your instrument model. Each can be independently set to one of the serial trigger types. The serial parameters related to the trigger are broken into two sections: Trigger:A:SERIAL xxx, consisting of parameters the user will change frequently, and BUS:B1:xxx, consisting of parameters the user will specify once (bus definition).

**Conditions**

Requires a DPO4AUTO, DPO4EMBD, or DPO4COMP application module.

**Group**

Trigger

**Syntax**

```
TRIGger:A:BUS {I2C|SPI|CAN|RS232|PARallel}
TRIGger:A:BUS?
```

**Arguments**

- **I2C** specifies the Inter-IC bus.
- **SPI** specifies the Serial Peripheral Interface bus.
- **CAN** specifies the Controller Area Network bus.
- **PARallel** specifies the Parallel bus.

*NOTE.* Parallel bus commands work with MSO4000 Series oscilloscopes only.

**TRIGger:A:BUS:B<x>:AUDio:CONDition**

Sets or returns the trigger condition for the AUDIO bus: Start of Frame or Data.

**Conditions**

This command requires DPO4AUDIO application module.

**Group**

Trigger

**Syntax**

```
TRIGger:A:BUS:B<x>:AUDio:CONDition {SOF|DATA}
TRIGger:A:BUS:B<x>:AUDio:CONDition?
```

**Arguments**

- **SOF** enables triggering on the start of frame.
- **DATA** enables triggering on matching data.
**TRIGger:A:BUS:B<x>:AUDio:DATa:HIVALue**

Sets or returns the trigger data upper word for the AUDIO bus.

**Conditions**
This command requires DPO4AUDIO application module.

**Group**
Trigger

**Syntax**

```
TRIGger:A:BUS:B<x>:AUDio:DATa:HIVALue <String>
TRIGger:A:BUS:B<x>:AUDio:DATa:HIVALue?
```

**Arguments**

<NR1> is the trigger data offset value.

**TRIGger:A:BUS:B<x>:AUDio:DATa:OFFSet**

Sets or returns the trigger data offset for the AUDIO bus.

**Conditions**
This command requires DPO4AUDIO application module.

**Group**
Trigger

**Syntax**

```
TRIGger:A:BUS:B<x>:AUDio:DATa:OFFSet <NR1>
TRIGger:A:BUS:B<x>:AUDio:DATa:OFFSet?
```

**Arguments**

<NR1> is the trigger data offset value.

**TRIGger:A:BUS:B<x>:AUDio:DATa:QUALifier**

Sets or returns the trigger data qualifier for the AUDIO bus.

**Conditions**
This command requires DPO4AUDIO application module.

**Group**
Trigger

**Syntax**

```
TRIGger:A:BUS:B<x>:AUDio:DATa:QUALifier {LESSThan|MOREThan|EQUAL|UNEQual|LESSEqual|MOREREQual|INrange|OUTrange}
TRIGger:A:BUS:B<x>:AUDio:DATa:QUALifier?
```
Arguments

- LESSThan sets the Audio data qualifier to less than.
- MOREThan sets the Audio data qualifier to greater than.
- EQUa1 sets the Audio data qualifier to equal.
- UNEQua1 sets the Audio data qualifier to not equal.
- LESSEQua1 sets the Audio data qualifier to less than or equal.
- MOREEQua1 sets the Audio data qualifier to greater than or equal.
- INrange sets the Audio data qualifier to in range.
- OUTrange sets the Audio data qualifier to out of range.

**TRIgger:A:BUS:B<x>:AUDio:DATa:VALue**

Sets or returns the trigger data lower word for the AUDIO bus.

**Conditions**

This command requires DPO4AUDIO application module.

**Group**

Trigger

**Syntax**

TRIgger:A:BUS:B<x>:AUDio:DATa:VALue <String>
TRIgger:A:BUS:B<x>:AUDio:DATa:VALue?

**Arguments**

<String> specifies the trigger data lower word.

**TRIgger:A:BUS:B<x>:AUDio:DATa:WORD**

Sets or returns the trigger data alignment for the AUDIO bus.

**Conditions**

This command requires DPO4AUDIO application module.

**Group**

Trigger

**Syntax**

TRIgger:A:BUS:B<x>:AUDio:DATa:WORD {EITher|LEFt|RIGht}
TRIgger:A: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.

**TRIGger:A:BUS:B<x>:CAN:CONDition**

Sets or returns the CAN trigger condition for bus \(<x>\), where \(x\) is the bus number.

**Conditions**
Requires a DPO4AUTO application module.

**Group**
Trigger

**Syntax**

```
TRIGger:A:BUS:B<x>:CAN:CONDition
{SOF|FRAME|IDentifier|DATA|IDANDDATA|EOF|ACKMISS}
TRIGger:A:BUS:B<x>:CAN:CONDition?
```

**Arguments**

- **SOF** enables triggering on the start of frame.
- **FRAME** 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.
- **ACKMISS** enables triggering on a missing acknowledge.

**Examples**

```
CAN:CONDITION EOF indicating an end of file condition.

TRIGGER:A:BUS:B1:CAN:CONDITION DATA enables triggering on matching
CAN data.
```

**TRIGger:A:BUS:B<x>:CAN:DATa:DIRection**

Sets or returns the CAN trigger data direction to be valid on a Read, Write, or Either condition for bus \(<x>\), where \(x\) is the bus number. This applies only, if the trigger condition is ID.

**Conditions**
Requires a DPO4AUTO application module.

**Group**
Trigger
Commands Listed in Alphabetical Order

**Syntax**

TRIGger:A:BUS:B<x>:CAN:DATA:DIRection {READ|WRITE|NOCARE}

TRIGger:A:BUS:B<x>:CAN:DATA:DIRection?

**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:BUS:B<x>:CAN:DATa:QUALifier**

Sets or returns the CAN data qualifier for bus <x>, where x is the bus number. This applies only, if the trigger condition is IDANDDATA or DATA.

**Conditions**

Requires a DPO4AUTO application module.

**Group**

Trigger

**Syntax**

TRIGger:A:BUS:B<x>:CAN:DATA:QUALifier

{LESSThan|Than|EQUal|UNEQual|LESSEQual|EQual}

TRIGger:A:BUS:B<x>:CAN:DATA:QUALifier?

**Arguments**

LESSThan sets the oscilloscope to trigger when the data is less than the qualifier value.

Than sets the oscilloscope to trigger when the data is 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.

EQual sets the oscilloscope to trigger when the data is 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 THAN, indicating that the
oscilloscope is set to trigger when the data is than the qualifier value.

**TRIGger:A:BUS:B<x>:CAN:DATa:SIZe**

Sets or returns the length of the data string in bytes for a CAN trigger if the
condition is DATA or IDANDDATA. Applies to bus <x>, where x is the bus
number.

**Conditions**

This command requires a DPO4AUTO application module.

**Group**

Trigger

**Syntax**

TRIGger:A:BUS:B<x>:CAN:DATa:SIZe <NR1>
TRIGger:A:BUS:B<x>:CAN:DATa:SIZe?

**Arguments**

<NR1> is the length of the data string in bytes.

**TRIGger:A:BUS:B<x>:CAN:DATa:VALue**

Sets or returns the binary data string to be used for a CAN trigger if the trigger
condition is ID or IDANDDATA. Applies to bus <x>, where x is the bus number.

**Conditions**

This command requires a DPO4AUTO application module.

**Group**

Trigger

**Syntax**

TRIGger:A:BUS:B<x>:CAN:DATa:VALue <QString>
TRIGger:A:BUS:B<x>:CAN:DATa:VALue?

**Arguments**

<QString> is the CAN data value in binary format. The only allowed characters
in the QString are 0, 1, and X.

**Examples**

**TRIGger:A:BUS:B<x>:CAN:FRAMEtype**

Sets or returns the frame type for a CAN FRAMETYPE trigger. Applies to bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4AUTO application module. This command is only valid when the TRIGger:A:BUS:B<x>:CAN:CONDITION is FRAMETYPE.

**Group**
Trigger

**Syntax**
TRIGger:A:BUS:B<x>:CAN:FRAMEtype
{DATA|REMoTe|ERRor|OVERLoad}
TRIGger:A: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:BUS:B<x>:CAN{:IDentifier|:ADDRess}:MODe**

Sets or returns the CAN addressing mode for bus <x>, where x is the bus number. Use this command to do the following:
- Trigger on ID
- Trigger in IDANDDATA

**Conditions**
This command requires a DPO4AUTO application module.

**Group**
Trigger

**Syntax**
TRIGger:A:BUS:B<x>:CAN{:IDentifier|:ADDRess}:MODe
{STandard|EXTended}
TRIGger:A:BUS:B<x>:CAN{:IDentifier|:ADDRess}:MODe?
Arguments

STandard specifies the standard addressing mode.
EXTended specifies the extended addressing mode.

**TRIGger:A:BUS:B<x>:CAN{:IDenti|:ADDRess}:VALue**

Sets or returns the binary address string used for a CAN trigger if the trigger condition is ID or IDANDDATA. Applies to bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4AUTO application module.

**Group**
Trigger

**Syntax**

```
TRIGger:A:BUS:B<x>:CAN{:IDenti|:ADDRess}:VALue <QString>
TRIGger:A:BUS:B<x>:CAN{:IDenti|:ADDRess}:VALue?
```

**Arguments**

- `<QString>` is up to 29 bits specifying the binary CAN identifier value. The only allowed characters in the QString are 0, 1, and X.

**Examples**

```
```

**TRIGger:A:BUS:B<x>:FLEXray:CONDition**

Sets or returns the trigger condition for FLEXRAY.

**Group**
Trigger

**Syntax**

```
TRIGger:A:BUS:B<x>:FLEXray:CONDition {SOF|FRAMEType|IDenti|CYCLEcount|HEADER|DATA|IDANDDATA|EOF|ERROR}
TRIGger:A:BUS:B<x>:FLEXray:CONDition?
```

**Arguments**

- **SOF** sets the FLEXRAY trigger condition to start of frame.
- **FRAMEType** sets the FLEXRAY trigger condition to frame type.
- **IDenti** sets the FLEXRAY trigger condition to identifier.
- **CYCLEcount** sets the FLEXRAY trigger condition to cycle count.
- **HEADER** sets the FLEXRAY trigger condition to header.
Commands Listed in Alphabetical Order

DATA sets the FLEXRAY trigger condition to data.
IDANDDATA sets the FLEXRAY trigger condition to id and data.
EOF sets the FLEXRAY trigger condition to end of frame.
ERROR sets the FLEXRAY trigger condition to error.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:CONDITION SOF indicating the
GLEXRAY condition is start of frame.

TRIGGER:A:BUS:B1:FLEXRAY:CONDITION SOF indicating the
GLEXRAY condition is start of frame.

TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:HIVALue

Sets or returns the binary data string to be used for FLEXRAY cycle count high value.

Group Trigger

Syntax

TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:HIVALue <QString>
TRIGger:A: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:BUS:B1:FLEXRAY:CYCLECOUNT:HIVALUE "XXXXXX" indicating
the cycle count high value is don’t care.

TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:QUALifier

Sets or returns the FLEXRAY cycle count qualifier.

Group Trigger
Commands Listed in Alphabatical Order

**Syntax**

```
TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:QUALifier
{LESSThan|MOREThan|EQUal|UNEQual|LESSEqual|MORREEQual|
INrange|OUTrange}
TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:QUALifier?
```

**Arguments**

- **LESSThan** sets the FLEXRAY cycle count qualifier to less than.
- **MOREThan** sets the FLEXRAY cycle count qualifier to more than.
- **EQUal** sets the FLEXRAY cycle count qualifier to equal.
- **UNEQual** sets the FLEXRAY cycle count qualifier to not equal.
- **LESSEqual** sets the FLEXRAY cycle count qualifier to less than or equal.
- **MORREEQual** sets the FLEXRAY cycle count qualifier to greater than or equal.
- **INrange** sets the FLEXRAY cycle count qualifier to in range.
- **OUTrange** sets the FLEXRAY cycle count qualifier to out of range.

**Examples**

```
TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:QUALIFIER LESSTHAN sets the
cycle count qualifier to LESSTHAN.

that the cycle count qualifier is set to EQUAL.
```

**TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:VALue**

Sets or returns the binary data string to be used for FLEXRAY cycle count low value.

**Group**

Trigger

**Syntax**

```
TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:VALue <QString>
TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:VALue?
```

**Arguments**

- `<QString>` is a quoted string that is the FLEXRAY binary data string that is the cycle count low value.

**Examples**

```
TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:VALUE “11001101” sets the
cycle count value to 11001101.
```
TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:VALUE "XXXXXX" indicating the
cycle count value is don’t care.

TRIGGER:A:BUS:B<x>:FLEXray:DATa:HIVALue

Sets or returns the high binary data string to be used for FLEXRAY trigger if
trigger condition is ID or IDANDDATA.

Group
Trigger

Syntax
TRIGGER:A:BUS:B<x>:FLEXray:DATa:HIVALue <QString>
TRIGGER:A:BUS:B<x>:FLEXray:DATa:HIVALue?

Arguments
<QString> is a quoted string that is the FLEXRAY binary data high value.

Examples
"11001101XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXX1" sets the binary data string high value to
"11001101XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXX1".

"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXX" indicating the binary data string high value is don’t care.

TRIGGER:A:BUS:B<x>:FLEXray:DATa:OFFSet

Sets or returns the offset of the data string in bytes to be used for FLEXRAY trigger.

Group
Trigger

Syntax
TRIGGER:A:BUS:B<x>:FLEXray:DATa:OFFSet <NR1>
TRIGGER:A: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 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:BUS:B1:FLEXRAY:DATA:OFFSET 0 indicating that a data offset  
of 0.

**TRIGger:A:BUS:B<x>:FLEXray:DATa:QUALifier**

Sets or returns the FLEXRAY data qualifier.

**Group**  
Trigger

**Syntax**  
TRIGger:A:BUS:B<x>:FLEXray:DATa:QUALifier  
{LESSThan|MOREThan|EQUal|UNEQual|LESSEQual|MOR EEQual|  
INrange|OUTrange}  
TRIGger:A:BUS:B<x>:FLEXray:DATa:QUALifier?

**Arguments**  
LESSThan sets the FLEXRAY data qualifier to less than.  
MOREThan sets the FLEXRAY data qualifier to greater than.  
EQUal sets the FLEXRAY data qualifier to equation.  
UNEQual sets the FLEXRAY data qualifier to not equal.  
LESSEQual sets the FLEXRAY data qualifier to less than or equal.  
MOREEQual sets the FLEXRAY data qualifier to greater than or equal.  
INrange sets the FLEXRAY data qualifier to in range.  
OUTrange sets the FLEXRAY data qualifier to out of range.

**Examples**  
TRIGGER:A:BUS:B1:FLEXRAY:DATA:QUALIFIER LESSTHAN sets the data  
qualifier to LESSTHAN.  
TRIGGER:A:BUS:B1:FLEXRAY:DATA:QUALIFIER EQUAL indicating the data  
qualifier is EQUAL.

**TRIGger:A:BUS:B<x>:FLEXray:DATa:SIZe**

Sets or returns the length of the data string in bytes to be used for FLEXRAY  
trigger.
Commands Listed in Alphabetical Order

**Group** Trigger

**Syntax**

TRIGger:A:BUS:B<x>:FLEXray:DATa:SIZE <NR1>
TRIGger:A: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:B<x>:FLEXray:DATa:VALue**

Sets or returns the low binary data string to be used for FLEXRAY trigger condition if trigger condition is ID or IDANDDATA.

**Group** Trigger

**Syntax**

TRIGger:A:BUS:B<x>:FLEXray:DATa:VALue <QString>
TRIGger:A:BUS:B<x>:FLEXray:DATa:VALue?

**Arguments**

<QString> is a quoted string that is the binary data string for a FLEXRAY trigger if the trigger condition is ID or IDANDDATA.

**Examples**


**TRIGger:A:BUS:B<x>:FLEXray:EOFTYPE**

Sets or returns the end of file type used for FLEXRAY trigger.

**Group** Trigger
Syntax

```
TRIGGER:A:BUS:B<x>:FLEXray:EOFTYPE {STATIC|DYNAMIC|ANY}
TRIGGER:A: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:B<x>:FLEXray:ERRTYPE**

Sets or returns the error type be used for FLEXRAY trigger.

Group Trigger

Syntax

```
TRIGger:A:BUS:B<x>:FLEXray:ERRTYPE {CRCHeader|CRCTrailer|SYNCFrame|STARTupnosync|NULLFRStatic|NULLFRDynamic}
TRIGger:A:BUS:B<x>:FLEXray:ERRTYPE?
```

Arguments

- **CRCHeader** sets the FLEXRAY error type for a trigger to CRCHeader.
- **CRCTrailer** sets the FLEXRAY error type for a trigger to CRCTrailer.
- **SYNCFrame** sets the FLEXRAY error type for a trigger to SYNCFrame.
- **STARTupnosync** sets the FLEXRAY error type for a trigger to STARTupnosync.
- **NULLFRStatic** sets the FLEXRAY error type for a trigger to NULLFRStatic.
- **NULLFRDynamic** sets the FLEXRAY error type for a trigger to NULLFRDynamic.

Examples

```
TRIGGER:A:BUS:B1:FLEXRAY:ERRTYPE SYNCFRAME sets the FLEXRAY trigger type is SYNCFRAME.
```
TRIGger:A:BUS:B<x>:FLEXray:FRAMEID:HIVALue

Sets or returns the binary data string to be used for FLEXRAY frame ID high value.

Group
Trigger

Syntax
TRIGger:A:BUS:B<x>:FLEXray:FRAMEID:HIVALue <QString>
TRIGger:A:BUS:B<x>:FLEXray:FRAMEID:HIVALue?

Arguments
<QString> is a quoted string that is the binary frame id high value.

Examples


TRIGger:A:BUS:B<x>:FLEXray:FRAMEID:QUALifier

Sets or returns the FLEXRAY frame ID qualifier.

Group
Trigger

Syntax
TRIGger:A:BUS:B<x>:FLEXray:FRAMEID:QUALifier {LESSThan|MOREThan|EQUal|UNEQual|LESSEqual|MOREEQual|INrange|OUTrange}
TRIGger:A: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.
MOREEQual 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:BUS:B1:FLEXRAY:FRAMEID:QUALIFIER LESSTHAN sets the frame id qualifier to less than.


TRIGGER:A:BUS:B<x>:FLEXray:FRAMEID:VALue

Sets or returns the binary data string to be used for FLEXRAY frame ID low value.

Group

Trigger

Syntax

TRIGGER:A:BUS:B<x>:FLEXray:FRAMEID:VALUE <QString>
TRIGGER:A:BUS:B<x>:FLEXray:FRAMEID:VALUE?

Arguments

<QString> is a quoted string that is the FLEXRAY frame ID low value.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:VALUE "11001100101" sets the frame id value to 11001100101.


TRIGGER:A:BUS:B<x>:FLEXray:FRAMEType

Sets or returns the frame type for FLEXRAY.

Group

Trigger

Syntax

TRIGGER:A:BUS:B<x>:FLEXray:FRAMEType \{NORMAL|PAYLoad|NULL|SYNC|STARTup\}
TRIGGER:A: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:BUS:B<x>:FLEXray:HEADER:CRC**

Sets or returns the CRC portion of the binary header string to be used for FLEXRAY trigger.

**Group**

Trigger

**Syntax**

TRIGger:A:BUS:B<x>:FLEXray:HEADER:CRC <QString>

TRIGger:A: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 "XXXXXXXXXX" indicating the CRC portion of the binary header string is don’t care.

**TRIGger:A:BUS:B<x>:FLEXray:HEADER:CYCLEcount**

Sets or returns the cycle count portion of the binary header string to be used for FLEXRAY trigger.

**Group**

Trigger

**Syntax**

TRIGger:A:BUS:B<x>:FLEXray:HEADER:CYCLEcount <QString>

TRIGger:A: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:BUS:B1:FLEXRAY:HEADER:CYCLECOUNT "XXXXXXXX" indicating the cycle count is don’t care.

TRIGGER:A:BUS:B<x>:FLEXray:HEADER:FRAMEID

Sets or returns the frame id portion of the binary header string to be used for FLEXRAY trigger.

Group

Trigger

Syntax

TRIGGER:A:BUS:B<x>:FLEXray:HEADER:FRAMEID <QString>
TRIGGER:A:BUS:B<x>:FLEXray:HEADER:FRAMEID?

Arguments

<QString> is a quoted string that is 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:BUS:B1:FLEXRAY:HEADER:FRAMEID "XXXXXXXXXXX" indicating the frame id portion of the binary header string are don’t cares.

TRIGGER:A:BUS:B<x>:FLEXray:HEADER:INDBits

Sets or returns the indicator bits portion of the binary header string to be used for FLEXRAY trigger.

Group

Trigger

Syntax

TRIGGER:A:BUS:B<x>:FLEXray:HEADER:INDBits <QString>
TRIGGER:A: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:BUS:B1:FLEXRAY:HEADER:INDBITS "XXXXX" indicating that the indicator bits portion of the header string are don’t cares.

**TRIGger:A:BUS:B<x>:FLEXray:HEADER:PAYLength**

Sets or returns the payload length portion of the binary header string to be used for FLEXRAY trigger.

**Group**

Trigger

**Syntax**

TRIGger:A:BUS:B<x>:FLEXray:HEADER:PAYLength <QString>

TRIGger:A:BUS:B<x>:FLEXray:HEADER:PAYLength?

**Arguments**

<QString> is the length of the payload portion of the Binary header string.

**Examples**


TRIGGER:A:BUS:B1:FLEXRAY:HEADER:PAYLENGTH "XXXXXXX" indicating the FLEXRAY header paylength is don’t care.

**TRIGger:A:BUS:B<x>:I2C:ADDRess:MODE**

Sets or returns the I2C address mode to 7 or 10-bit. Applies to bus <x>, where x is the bus number.

**Conditions**

This command requires a DPO4EMBD application module.

**Group**

Trigger

**Syntax**

TRIGger:A:BUS:B<x>:I2C:ADDRess:MODE {ADDR7|ADDR10}

TRIGger:A: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:BUS:B<x>:I2C:ADDRESS:TYPE

Sets or returns the I2C address type. The only supported address type is USER. Applies to bus <x>, where x is the bus number.

Conditions
This command requires a DPO4EMBD application module.

Group
Trigger

Syntax
TRIGger:A:BUS:B<x>:I2C:ADDRESS:TYPE {GENeralcall|STARtbyte|HSmode|EEPROM|USER}
TRIGger:A:BUS:B<x>:I2C:ADDRESS:TYPE?

Arguments
GENeralcall specifies a general call address.
STARtbyte specifies a start byte address.
HSmode specifies a high-speed mode address.
EEPROM specifies an EEPROM address.
USER specifies a user address.

TRIGger:A:BUS:B<x>:I2C:ADDRESS:VALUE

Sets or returns the binary address string used for the I2C trigger if the trigger condition is ADDRESS or ADDRANDDATA. Applies to bus <x>, where x is the bus number.

Conditions
This command requires a DPO4EMBD application module.

Group
Trigger

Syntax
TRIGger:A:BUS:B<x>:I2C:ADDRESS:VALUE <QString>
TRIGger:A: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:B<x>:I2C:CONDITION**

Sets or returns the trigger condition for an I2C trigger. Applies to bus <x>, where x is the bus number.

**Conditions**

This command requires a DPO4EMBD application module.

**Group**

Trigger

**Syntax**

TRIGger:A:BUS:B<x>:I2C:CONDITION

{START|STOP|REPEATstart|ACKMISS|ADDRESS|DATA|ADDRANDDATA}

TRIGger:A: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:BUS:B<x>:I2C:DATa:DIRection**

Sets or returns 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. Applies to bus <x>, where x is the bus number.

**Conditions**

This command requires a DPO4EMBD application module.

**Group**

Trigger
Commands Listed in Alphabetical Order

Syntax

TRIGger:A:BUS:B<x>:I2C:DATa:DIRection {READ|WRITE|NOCARE}
TRIGger:A: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:BUS:B<x>:I2C:DATa:SIZe

Sets or returns 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 x is the bus number.

Conditions

This command requires a DPO4EMBD application module.

Group

Trigger

Syntax

TRIGger:A:BUS:B<x>:I2C:DATa:SIZe <NR1>
TRIGger:A:BUS:B<x>:I2C:DATa:SIZe?

Arguments

<NR1> is the length of the data string in bytes.

TRIGger:A:BUS:B<x>:I2C:DATa:VALue

Sets or returns the binary data string used for I2C triggering if the trigger condition is DATA or ADDRANDDATA. Applies to bus <x>, where x is the bus number.

Conditions

This command requires a DPO4EMBD application module.

Group

Trigger

Syntax

TRIGger:A:BUS:B<x>:I2C:DATa:VALue <QString>
TRIGger:A: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.

TRIGger:A:BUS:B<x>:LIN:CONDition

Sets or returns the trigger condition for LIN.

Group

Trigger

Syntax

TRIGger:A:BUS:B<x>:LIN:CONDition {SYNCFIELD|IDENTIFIER|DATA|IDANDDATA|WAKEUp|SLEEP|ERROR}

TRIGger:A: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:BUS:B1:LIN:CONDITION SYNCFIELD indicating the LIN trigger condition is sync field.

TRIGger:A:BUS:B<x>:LIN:DATa:HIVALue

Sets or returns the binary data string to be used for LIN trigger if trigger condition is ID or IDANDDATA.

Group

Trigger

Syntax

TRIGger:A:BUS:B<x>:LIN:DATa:HIVALue <QString>

TRIGger:A: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:BUS:B1:LIN:DATA:HIVALUE "xxxxxxxx" indicating the high value is don't care.

TRIGger:A:BUS:B<x>:LIN:DATa:QUALifier

Sets or returns the LIN data qualifier. This only applies if the trigger condition is IDANDDATA or DATA.

Group

Trigger

Syntax

TRIGger:A:BUS:B<x>:LIN:DATa:QUALifier

{LESSThan|MOREThan|EQUAL|UNEQual|LESSEqual|MOREEQual|INrange|OUTrange}

TRIGger:A: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.
MOREEQual 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:BUS:B1:LIN:DATA:QUALIFIER OUTrange sets the data qualifier to out of range.


TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER EQUAL indicating the data qualifier is set to equal.
**TRIGger:A:BUS:B<x>:LIN:DATa:SIZe**

Sets or returns the length of the data string in bytes to be used for LIN trigger.

**Group**
Trigger

**Syntax**
TRIGger:A:BUS:B<x>:LIN:DATa:SIZe <NR1>
TRIGger:A: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:BUS:B1:LIN:DATA:SIZE 1 indicating the data size is 1 byte.

**TRIGger:A:BUS:B<x>:LIN:DATa:VALue**

Sets or returns the binary data string to be used for LIN trigger condition if trigger condition is ID or IDANDDATA.

**Group**
Trigger

**Syntax**
TRIGger:A:BUS:B<x>:LIN:DATa:VALue <QString>
TRIGger:A: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:BUS:B<x>:LIN:ERRTYPE**

Sets or returns the error type be used for LIN trigger.

**Group**
Trigger
**Syntax**

```
TRIGger:A:BUS:B<x>:LIN:ERRTYPE
{SYNC|PARity|CHecksum|HEADertime|RESPtime|FRAMetime}
TRIGger:A: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.
- **HEADertime** sets the LIN error type to header time.
- **RESPtime** sets the LIN error type to response time.
- **FRAMetime** sets the LIN error type to frame time.

**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:BUS:B<x>:PARallel:VALue**

Sets or returns the binary data string to be used for a Parallel trigger. Applies to bus \(<x>\), where \(x\) is the bus number.

**Group**

Trigger

**Syntax**

```
TRIGger:A:BUS:B<x>:PARallel:VALue <QString>
TRIGger:A:BUS:B<x>:PARallel:VALue?
```

**Arguments**

\(<QString>\) is the binary data string.

**TRIGger:A:BUS:B<x>:RS232C:CONDition**

Sets or returns the condition for an RS-232C trigger, where \(x\) is the bus number.

**Conditions**

This command requires a DPO4COMP application module.

**Group**

Trigger

**Syntax**

```
TRIGger:A:BUS:B<x>:RS232C:CONDition {RXSTArt|RXDATA|RXENDPacket|TXSTArt|TXDATA|TXENDPacket}
TRIGger:A:BUS:B<x>:RS232C:CONDition?
```

**Arguments**

- **RXSTArt** specifies a search based on the RX Start Bit.
- **RXDATA** specifies a search based on RX Data.
- **RXENDPacket** specifies a search based on the RX End of Packet condition.
- **TXSTArt** specifies a search base on the TX Start Bit.
- **TXDATA** specifies a search based on TX Data.
- **TXENDPacket** specifies a search based on the TX End of Packet condition.

**TRIGger:A:BUS:B<x>:RS232C:RX:DATa:SIZe**

Sets or returns the length of the data string in Bytes for an RS-232 Trigger if the trigger condition is RXDATA. Applies to bus \(<x>\), where \(x\) is the bus number.

**Conditions**

This command requires a DPO4COMP application module.
Commands Listed in Alphabetical Order

**Group**  Trigger

**Syntax**  
TRIGger:A:BUS:B<x>:RS232C:RX:DATa:SIZe <NR1>  
TRIGger:A:BUS:B<x>:RS232C:RX:DATa:SIZe?

**Arguments**  
<NR1> is the length of the data string in bytes.

**TRIGger:A:BUS:B<x>:RS232C:RX:DATa:VALue**

Sets or returns the binary data string for an RS-232 trigger if the trigger condition involves RX. Applies to bus <x>, where x is the bus number.

**Conditions**  
This command requires a DPO4COMP application module.

**Group**  Trigger

**Syntax**  
TRIGger:A:BUS:B<x>:RS232C:RX:DATa:VALue?

**Arguments**  
<Qstring> is the binary data string to be used for the trigger.

**TRIGger:A:BUS:B<x>:RS232C:TX:DATa:SIZe**

Sets or returns the length of the data string for an RS-232 trigger if the trigger condition is TXDATA. Applies to bus <x>, where x is the bus number.

**Conditions**  
This command requires a DPO4COMP application module.

**Group**  Trigger

**Syntax**  
TRIGger:A:BUS:B<x>:RS232C:TX:DATa:SIZe <NR1>  
TRIGger:A:BUS:B<x>:RS232C:TX:DATa:SIZe?

**Arguments**  
<NR1> is the length of the data string in Bytes.
**TRIGger:A:BUS:B<x>:RS232C:TX:DATa:VALue**

Sets or returns the binary data string for an RS-232 trigger if the condition involves TX. Applies to bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4COMP application module.

**Group**
Trigger

**Syntax**
TRIGger:A:BUS:B<x>:RS232C:TX:DATa:VALue
TRIGger:A:BUS:B<x>:RS232C:TX:DATa:VALue?

**Arguments**
<Qstring> is the binary data string to be used for the trigger.

**TRIGger:A:BUS:B<x>:SPI:CONDition**

Sets or returns the trigger condition for a SPI trigger. Applies to bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4EMBD application module.

**Group**
Trigger

**Syntax**
TRIGger:A:BUS:B<x>:SPI:CONDition
{SS|STARTofframe|MISO|MOSI|MISOMOSI}
TRIGger:A:BUS:B<x>:SPI:CONDition?

**Related Commands**
BUS:B<x>:SPI:IDLETime, BUS:B<x>:SPI:FRAMING

**Arguments**
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.

MISO specifies the Master-In Slave-Out condition.

MOSI specifies the Master-Out Slave-In condition.

MISOMOSI specifies the Master-In Slave-Out and Master-Out Slave-In conditions.
**TRIGger:A:BUS:B<x>:SPI:DATa{:IN|:MISO}:VALue**

Sets or returns the binary data string to be used for a SPI trigger if the trigger condition is MISO or MISOMOSI. Applies to bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4EMBD application module.

**Group**
Trigger

**Syntax**
```
TRIGger:A:BUS:B<x>:SPI:DATa{:IN|:MISO}:VALue <QString>
TRIGger:A:BUS:B<x>:SPI:DATa{:IN|:MISO}: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.

**TRIGger:A:BUS:B<x>:SPI:DATa{:OUT|:MOSI}:VALue**

Sets or returns the binary data string to be used for a SPI trigger if the trigger condition is MOSI or MISOMOSI. Applies to bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4EMBD application module.

**Group**
Trigger

**Syntax**
```
TRIGger:A:BUS:B<x>:SPI:DATa{:OUT|:MOSI}:VALue <QString>
TRIGger:A:BUS:B<x>:SPI:DATa{:OUT|:MOSI}:VALue?
```

**Arguments**
<QString> is the binary data string with the number of bits specified by the TRIGger:A:BUS:B<x>:SPI:DATa:SIZE command. The only allowed characters in the QString are 0, 1, and X.

**TRIGger:A:BUS:B<x>:SPI:DATa:SIZE**

Sets or returns the length of the data string to be used for a SPI trigger if the trigger condition is MISO, MOSI, or MISOMOSI. Applies to bus <x>, where x is the bus number.

**Conditions**
This command requires a DPO4EMBD application module.
Commands Listed in Alphabetical Order

**Group**  
Trigger

**Syntax**  
TRIGger:A:BUS:B<x>:SPI:DATa:SIZe <NR1>  
TRIGger:A:BUS:B<x>:SPI:DATa:SIZe?

**Arguments**  
<NR1> is the length of the data string in bytes.

**TRIGger:A:BUS:B<x>:USB:ADDRess:HIVALue**

Sets or returns 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:BUS:B<x>:USB:ADDRess:VALue to set the lower limit.

**Conditions**  
This command requires the DPO4USB application module.

**Group**  
Trigger

**Syntax**  
TRIGger:A:BUS:B<x>:USB:ADDRess:HIVALue <QString>  
TRIGger:A:BUS:B<x>:USB:ADDRess:HIVALue?

**Related Commands**  

**Arguments**  
<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).  

**TRIGger:A:BUS:B<x>:USB:ADDRess:VALue**

Sets or returns the binary address string to be used for USB trigger.  
This command also sets or returns the binary address string for the lower limit for inside-of-range and outside-of-range qualifiers for the USB trigger.

**Conditions**  
This command requires the DPO4USB application module.
## TRIGger:A:BUS:B<x>:USB:ADDRESS:VALUE

### Syntax

```
TRIGger:A:BUS:B<x>:USB:ADDRESS:VALUE <QString>
TRIGger:A:BUS:B<x>:USB:ADDRESS:VALUE?
```

### Related Commands

TRIGger:A:BUS:B<x>:USB:CONDition

### Arguments

<QString> within the range 0000000 to 1111111 (00 hex to 7F hex).

### Examples

```
```

Sets the binary address to 0001000 (08 hex).

```
```

Might return
```
```

Which indicates that the binary address is 100000 (40 hex).

## TRIGger:A:BUS:B<x>:USB:CONDition

Sets or returns the trigger condition for the USB trigger.

### Conditions

This command requires the DPO4USB application module.

### Group

Trigger

### Syntax

```
TRIGger:A:BUS:B<x>:USB:CONDition
{SYNC|RESET|SUSPEND|RESUME|EOP|TOKENPacket|DATAPacket|HANDSHAKEPacket|SPECIALPacket|ERRor}
TRIGger:A: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:BUS:B<x>:USB:DATa:HIVALue**

Sets or returns 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 DATAPacket. Use the command TRIGger:A:BUS:B<x>:USB:DATa:VALue to set the lower limit.

**Conditions**

This command requires the DPO4USB application module.

**Group**

Trigger

**Syntax**

TRIGger:A:BUS:B<x>:USB:DATa:HIVALue <QString>

TRIGger:A:BUS:B<x>:USB:DATa:HIVALue?

**Related Commands**


**Arguments**

<QString> within the range 00000000 to 11111111 (00 hex to FF hex).

**Examples**


**TRIGger:A:BUS:B<x>:USB:DATa:OFFSet**

Sets or returns the data offset in bytes to trigger on. The minimum and default values are 0 and the maximum is 1024.
**Conditions**

This command requires the DPO4USB application module.

**Group**

Trigger

**Syntax**

```
TRIGger:A:BUS:B<x>:USB:DATa:OFFSET <NR1>
TRIGger:A:BUS:B<x>:USB:DATa:OFFSET?
```

**Related Commands**

`TRIGger:A:BUS:B<x>:USB:CONDition`

**Arguments**

 `<NR1>` is number in the range 0 to 1024.

**Examples**

```
```

sets the data offset to 36.

```
:TRIGGER:A:BUS:B1:USB:DATA:OFFSET 0, indicating that the
data offset is the default value, 0.
```

**TRIGger:A:BUS:B<x>:USB:DATa:SIZe**

Sets or returns the number of contiguous data bytes to trigger on. The minimum and default values are 1 and maximum is 16.

**Conditions**

This command requires the DPO4USB application module.

**Group**

Trigger

**Syntax**

```
TRIGger:A:BUS:B<x>:USB:DATa:SIZe <NR1>
TRIGger:A:BUS:B<x>:USB:DATa:SIZe?
```

**Arguments**

 `<NR1>` is number in the range 1 to 16.

**Examples**

```
```

sets the oscilloscope to trigger on four contiguous data bytes.

```
:TRIGGER:A:BUS:B1:USB:DATA:SIZE 6 indicating that the oscilloscope will
trigger on 6 contiguous data bytes.
```
**TRIGger:A:BUS:B<x>:USB:DATa:TYPe**

Sets or returns the data type for when the trigger condition is set to DATAPacket.

**Conditions**
This command requires the DPO4USB application module.

**Group**
Trigger

**Syntax**
TRIGger:A:BUS:B<x>:USB:DATa:TYPe {ANY|DATA0|DATA1}

TRIGger:A:BUS:B<x>:USB:DATa:TYPe?

**Related Commands**
TRIGger:A: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.

**Examples**
TRIGGER:A:BUS:B1:USB:DATA:TYPE DATA0 sets the oscilloscope to trigger on a DATA0 data packet type.


**TRIGger:A:BUS:B<x>:USB:DATa:VALue**

Sets or returns the binary data string to be used for the USB trigger when the trigger condition is DATAPacket.

This command also sets or returns the binary data string for the lower limit for inside-of-range and outside-of-range qualifiers for the USB trigger when trigger condition is DATAPacket.

**Conditions**
This command requires the DPO4USB application module.

**Group**
Trigger

**Syntax**
TRIGger:A:BUS:B<x>:USB:DATa:VALue <QString>

TRIGger:A:BUS:B<x>:USB:DATa:VALue?
Commands Listed in Alphabetical Order

Related Commands
TRIGger:A:BUS:B<x>:USB:CONDition

Arguments
<String> within the range 00000000 to 11111111 (00 hex to FF hex).

Examples
TRIGGER:A:BUS:B1:USB:DATA:VALUE “00010000” sets the binary address to 00010000 (08 hex).

TRIGger:A:BUS:B<x>:USB:ENDPoint:VALue

Sets or returns the binary endpoint string to be used for the USB trigger.

Conditions
This command requires the DPO4USB application module.

Group
Trigger

Syntax
TRIGger:A:BUS:B<x>:USB:ENDPoint:VALue <QString>
TRIGger:A:BUS:B<x>:USB:ENDPoint:VALue?

Related Commands
TRIGger:A:BUS:B<x>:USB:CONDition

Arguments
<String> 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:B<x>:USB:ERRORTYPe

Sets or returns the error type used when the trigger condition is set to ERRor.

Conditions
This command requires the DPO4USB application module.
Commands Listed in Alphabetical Order

**Group** Trigger

**Syntax**

```
TRIGger:A:BUS:B<x>:USB:ERRORYPE {PID|CRC5|CRC16|BITSTUFFing}
TRIGger:A:BUS:B<x>:USB:ERRORYPE?
```

**Related Commands** TRIGger:A: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:CONDITION PID sets the error trigger condition to packet ID.
:TRIGGER:A:BUS:B1:USB:CONDITION PID, indicating the error trigger condition is packet ID.
```

**TRIGger:A:BUS:B<x>:USB:HANDSHAKEType**

Sets or returns the handshake type for the USB trigger.

**Conditions** This command requires the DPO4USB application module.

**Group** Trigger

**Syntax**

```
TRIGger:A:BUS:B<x>:USB:HANDSHAKEType {ANY|NAK|ACK|STALL}
TRIGger:A:BUS:B<x>:USB:HANDSHAKEType?
```

**Related Commands** TRIGger:A: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.

**Examples**

TRIGGER:A:BUS:B1:USB:HANDSHAKETYPE ACK sets the handshake type to acknowledgement.


**TRIGger:A:BUS:B<x>:USB:QUALifier**

Sets or returns the USB trigger qualifier for address, endpoint and data.

**Conditions**

This command requires the DPO4USB application module.

**Group**

Trigger

**Syntax**

TRIGger:A:BUS:B<x>:USB:QUALifier

{LESSThan|MOREThan|EQUal|UNEQual|LESSEQual|MOREEQual|INrange|OUTrange}

TRIGger:A:BUS:B<x>:USB:QUALifier?

**Related Commands**

TRIGger:A:BUS:B<x>:USB:CONDition

**Arguments**

LESSThan triggers on an input value that is less than a set value.

MOREThan triggers on an input value that is greater than a set value.

EQUal triggers on an input value that is equal to a set value.

UNEQual triggers on an input value that is not equal to a set value.

LESSEQual triggers on an input value that is less than or equal to a set value.

MOREEQual triggers on an input value that is more than or equal to a set value.

INrange triggers on an input value that is within a range set by two values.

OUTrange triggers on an input value that is outside of a range set by two values.

**Examples**

TRIGGER:A:BUS:B1:USB:QUALIFIER LESSThan sets the oscilloscope to trigger when an address, data, or endpoint value is less than a set value.
:TRIGGER:A:BUS:B1:USB:QUALIFIER EQUAL indicating that the trigger condition is to trigger when an address, data, or endpoint is equal to a set value.

TRIGGER:A:BUS:B<x>:USB:SOFFRAMENUMber

Sets or returns 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.

Conditions
This command requires the DPO4USB application module.

Group
Trigger

Syntax
TRIGGER:A:BUS:B<x>:USB:SOFFRAMENUMber <QString>
TRIGGER:A:BUS:B<x>:USB:SOFFRAMENUMber?

Related Commands
TRIGGER:A: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 "0000001000" sets the start of frame number to 0000001000 (008 hex).

:TRIGGER:A:BUS:B1:USB:SOFFRAMENUMBER "0000001001", which indicates that the start of frame number is 0000001001 (009 hex).

TRIGGER:A:BUS:B<x>:USB:SPECIALType

Sets or returns the packet ID (PID) for the special packet.

Conditions
This command requires the DPO4USB application module.

Group
Trigger

Syntax
TRIGGER:A:BUS:B<x>:USB:SPECIALType {ANY|PREamble|RESERVED}
TRIGGER:A:BUS:B<x>:USB:SPECIALType?
### TRIGger:A:BUS:B<x>:USB:SPECIALTYPE

Sets or returns the special packet type for the USB trigger.

**Arguments**
- **ANY** indicates any type of special packet.
- **PREamble** indicates a preamble special packet.
- **RESERVED** indicates a reserved special packet.

**Examples**
- `TRIGGER:A:BUS1:USB:SPECIALTYPE PREAMBLE` sets the special packet type to preamble.

### TRIGger:A:BUS:B<x>:USB:TOKENType

Sets or returns the token type for the USB trigger.

**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:SOUrce**

Sets or returns the source for a Serial bus trigger.

**Conditions**  
This command requires a DPO4AUTO or DPO4EMBD application module.

**Group**  
Trigger

**Syntax**  
TRIGger:A:BUS:SOUrce {B1|B2|B3|B4}  
TRIGger:A:BUS:SOUrce?

**Arguments**  
B1 specifies the Bus 1 source.  
B2 specifies the Bus 2 source.  
B3 specifies the Bus 3 source.  
B4 specifies the Bus 4 source.

**TRIGger:A:EDGE? (Query Only)**

Returns the trigger source, coupling, and slope for the A edge trigger.

**Group**  
Trigger

**Syntax**  
TRIGger:A:EDGE?

**Related Commands**  
TRIGger:A:PULse?, TRIGger:A:LOGIc?

**Examples**  

**TRIGger:A:EDGE:COUPling**

Sets or returns the type of coupling for the A edge trigger.

**Group**  
Trigger
Commands Listed in Alphabetical Order

**TRIGger:A:EDGE:COUPling**

Syntax

TRIGger:A:EDGE:COUPling {DC|HFRej|LFRej|NOISErej}

TRIGger:A:EDGE:COUPling?

Related Commands


Arguments

DC specifies DC trigger coupling, which passes all input signals to the trigger circuitry.

HFRej specifies high-frequency rejection coupling, which attenuates signals above 50 kHz before passing the signals to the trigger circuitry.

LFRej specifies low-frequency rejection coupling, which attenuates signals below 50 kHz before passing the signals to the trigger circuitry.

NOISErej specifies noise-rejection coupling, which provides stable triggering by increasing the trigger hysteresis. Increased hysteresis reduces the trigger sensitivity to noise but may 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:EDGE:SLOpe**

Sets or returns the slope for the A edge trigger.

Group

Trigger

Syntax

TRIGger:A:EDGE:SLOpe {RISe|FALL}

TRIGger:A:EDGE:SLOpe?

Related Commands


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.

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:EDGE:SOUrce**

Sets or returns the source for the A edge trigger.

**Group**  Trigger

**Syntax**  
TRIGger:A:EDGE:SOUrce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15|EXT|LINE|AUX}  
TRIGger:A:EDGE:SOUrce?

**Related Commands**  TRIGger:A:EDGE:SLOpe, TRIGger:A:EDGE:COUPling

**Arguments**  
CH1–CH4 specifies an analog input channel as the A edge trigger source.
D0–D15 specifies a digital channel as the source (MSO models only).
EXT specifies an external trigger using the Aux In connector located on the front panel of the oscilloscope.
LINE specifies the AC line as the trigger source.
AUX specifies the Auxiliary Input as the trigger source (if available on your oscilloscope).

**Examples**  
TRIGGER:A:EDGE:SOURCE CH1 sets channel 1 as the A edge trigger source.

**TRIGger:A:HOLDoff? (Query Only)**

Returns the A trigger holdoff parameters. These parameters specify the time period during which the trigger circuitry is not looking to generate a trigger event.

**Group**  Trigger

**Syntax**  
TRIGger:A:HOLDoff?

**Related Commands**  TRIGger:A:HOLDoff:TIMe
Examples  TRIGGER:A:HOLDOFF? might return :TRIGGER:A:HOLDOFF:TIME 900.0000E-09; BY DEFAULT indicating that the A edge trigger holdoff time (by default) is 900 ns.

TRIGGER:A:HOLDOFF:TIME

Sets or returns the A trigger holdoff time.

Group  Trigger

Syntax  TRIGGER:A:HOLDOFF:TIME <NR3>
        TRIGGER:A:HOLDOFF:TIME?

Arguments  <NR3> specifies the holdoff time in seconds. The range is from 20 ns through 8.0 s.

          TRIGGER:A:HOLDOFF:TIME 10 sets the A trigger holdoff time to 10 s.

TRIGGER:A:LEVEL

Sets or returns the trigger level for the A trigger.

Group  Trigger

Syntax  TRIGGER:A:LEVEL {ECL|TTL|<NR3>}
        TRIGGER:A:LEVEL?

Arguments  ECL specifies a preset ECL high level of –1.3V.
          TTL specifies a preset TTL high level of 1.4V.
          <NR3> specifies the trigger level in user units (usually volts).

Examples  TRIGGER:A:LEVEL? might return :TRIGGER:A:LEVEL 1.3000E+00 indicating that the A edge trigger is set to 1.3 V.
          TRIGGER:A:LEVEL TTL sets the A edge trigger to TTL high level, which is 1.4 V.
**TRIGger:A:LEVel:AUXin**

Sets or returns the trigger level for the AUXIN port.

**Group**

Trigger

**Syntax**

TRIGger:A:LEVel:AUXin {<NR3>|ECL|TTL}

TRIGger:A:LEVel:AUXin?

**Arguments**

<NR3> specifies the trigger level, in volts.

ECL specifies a preset ECL trigger level of –1.3V.

TTL specifies a preset TTL trigger level of 1.4V.

**Examples**

TRIGGER:A:LEVEL:AUXIN ECL sets the auxiliary input trigger level to -1.3 volts.

TRIGGER:A:LEVEL:AUXIN? might return TRIGGER:A:LEVEL:AUXIN 0.0E+0 indicating the auxiliary input trigger level is 0.0 volts.

**TRIGger:A:LEVel:CH<x>**

Sets or returns the trigger level for the specified channel. Each channel can have an independent level.

**Group**

Trigger

**Syntax**

TRIGger:A:LEVel:CH<x> {<NR3>|TTL|ECL}

TRIGger:A:LEVel:CH<x>?

**Arguments**

<NR3> specifies the trigger level in user units (usually volts).

TTL specifies a preset TTL high level of 1.4V.

ECL specifies a preset ECL high level of –1.3V.

**Examples**

TRIGGER:A:LEVEL:CH2? might return TRIGGER:A:LEVEL:CH2 1.3000E+0 indicating that the A edge trigger is set to 1.3 V for channel 2.

TRIGGER:A:LEVEL:CH3 TTL sets the A edge trigger to TTL high level for channel 3.
**TRIGger:A:LEVel:D<x>**

Sets or returns the trigger level for the specified digital channel <x>, where x is the channel number. Each digital channel can have an independent level.

**Group**  
Trigger

**Syntax**  
TRIGger:A:LEVel:D<x> {<NR3>|ECL|TTL}  
TRIGger:A:LEVel:D<x>?

**Arguments**  
ECL specifies a preset ECL high level of –1.3V.  
TTL specifies a preset TTL high level of 1.4V.  
<NR3> specifies the trigger level in volts.

**TRIGger:A:LOGIc? (Query Only)**

Returns all of the A logic trigger parameters.

**Group**  
Trigger

**Syntax**  
TRIGger:A:LOGIc?

**Related Commands**  
TRIGger:A:LOGIc:CLAss

**Examples**  
SETHOLD;FUNCTION AND;THRESHOLD:CH1 20.0000E-3;CH2 0.0000;CH3 0.0000;CH4 0.0000;:TRIGGER:A:LOGIC:INPUT:CH1 X;CH2 X;CH3 X;CH4 X;CLOCK:SOURCE NONE;EDGE RISE;:TRIGGER:A:LOGIC:PATTERN:INPUT:CH1 X;CH2 X;CH3 X;CH4 X;:TRIGGER:A:LOGIC:PATTERN:WHEN TRUE;WHEN:LESSLIMIT 4.0000E-9;LIMIT 4.0000E-9;:TRIGGER:A:LOGIC:PATTERN:DELTATIME 4.00000E-9

**TRIGger:A:LOGIc:CLAss**

Sets or returns the class of the Logic Trigger. This command is used in conjunction with the TRIGger:A:TYPe command.
**Group**  
Trigger

**Syntax**  
TRIGger:A:LOGIc:CLASS \{LOGIC|SETHold\}  
TRIGger:A:LOGIc:CLASS?

**Related Commands**  
TRIGger:A:TYPe, TRIGger:A:PULse:CLAss

**Arguments**  
LOGIC sets the oscilloscope to trigger on logical combinations of the channels.

When the TRIGger:A:LOGIc:INPut:CLOck:SOUrce is NONE, LOGIC sets the oscilloscope to trigger when the specified logical combinations of channels 1, 2, 3, and 4 are met on four-channel oscilloscopes. On two-channel oscilloscopes, only channel 1 and channel 2 are available.

When the TRIGger:A:LOGIc:INPut:CLOck:SOUrce is set to one of the channels, LOGIC sets the oscilloscope to trigger when the specified logical combinations of the remaining channels is true during a transition on the clock channel.

SETHold sets the oscilloscope to trigger on setup and hold violations between a data source and a clock source. Use one channel input as the clock signal and a second channel input as the data input. The clocking and data levels are used to determine if a clock or data transition has occurred.

**Examples**  

TRIGGER:A:LOGIC:CLASS LOGIC sets the trigger A logic class to LOGIc, which causes the oscilloscope to trigger when the specified logical combinations of channels 1, 2, 3, and 4 are met.

**TRIGger:A:LOGIc:FUNCTION**

Sets or returns the logical combination of the input channels for the A pattern and A state logic triggers.

**Group**  
Trigger

**Syntax**  
TRIGger:A:LOGIc:FUNCTION \{AND|NANd|NOR|OR\}  
TRIGger:A:LOGIc:FUNCTION?

**Related Commands**  
TRIGger:A:LOGIc:INPut:CH<x>
**Arguments**

AND specifies to trigger if all conditions are true.

NAND specifies to trigger if any of the conditions is false.

NOR specifies to trigger if all conditions are false.

OR specifies to trigger if any of the conditions is true.

**Examples**


which indicates that the oscilloscope will trigger if the AND logic conditions are false.

TRIGGER:A:LOGIC:FUNCTION AND sets the logical combination of channels to be true when all conditions are true.

---

**TRIGger:A:LOGIc:INPut? (Query Only)**

Returns the logic input values for all channels. If a clock channel is defined, it returns the clock source and edge.

**Group**

Trigger

**Syntax**

TRIGger:A:LOGIc:INPut?

**Examples**

TRIGGER:A:LOGIC:INPUT? might return :TRIGGER:A:LOGIC:INPUT:CH1 HIGH;CH2 X;CH3 X indicating that a logic high is expected on channel 1 while channel 2 and channel three are “don’t care.”

---

**TRIGger:A:LOGIc:INPut:CH<x>**

Sets or returns the logical input condition for the channel specified by <x>.

**Group**

Trigger

**Syntax**

TRIGger:A:LOGIc:INPut:CH<x> {HIGH|LOW|X}

TRIGger:A:LOGIc:INPut:CH<x>?  

**Arguments**

HIGH specifies the logic high.

LOW specifies the logic low.
X specifies a "don’t care" state.

**Examples**

TRIGGER:A:LOGIC:INPUT:CH1? might return :TRIGGER:LOGIC:INPUT:CH1 X indicating that the setting for the A logic trigger input to channel 1 does not matter.

TRIGGER:A:LOGIC:INPUT:CH2 HIGH sets the A logic trigger input to logic HIGH for channel 2.

---

**TRIGger:A:LOGIc:INPut:CLOCk:EDGE**

Sets the polarity of the clock channel.

**Group**

Trigger

**Syntax**

TRIGger:A:LOGIC:INPUT:CLOCK:EDGE {FALL|RISe}

TRIGger:A:LOGIC:INPUT:CLOCK:EDGE?

**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.

---

**TRIGger:A:LOGIc:INPut:CLOCk:SOUrce**

Sets or returns the channel to use as the clock source. The clock can be selected as NONE. A selection of None implies pattern trigger. Any other selection implies state trigger.

**Group**

Trigger

**Syntax**

TRIGger:A:LOGIC:INPUT:CLOCK:SOURce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15|NONE}

TRIGger:A:LOGIC:INPUT:CLOCK:SOURce?

**Arguments**

CH1–CH4 specifies the analog input channel source.

D0–D15 specifies the digital input channel source.

NONE specifies a Pattern trigger.
**TRIGger:A:LOGIc:INPut:D<x>**

Sets or returns the logic pattern for a trigger on digital channel <x>, where x is the channel number.

**Group** Trigger

**Syntax**

TRIGger:A:LOGIc:INPut:D<x> {HIGH|LOW|X}

TRIGger:A:LOGIc:INPut:D<x>?

**Arguments**

High specifies the logic high state.

Low specifies the logic low state.

X specifies a "don't care" state.

---

**TRIGger:A:LOGIc:PATtern? (Query Only)**

Returns the conditions used for generating an A logic pattern trigger, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger.

**Group** Trigger

**Syntax**

TRIGger:A:LOGIc:PATtern?

**Examples**

TRIGGER:A:LOGIC:PATTERN? might return


---

**TRIGger:A:LOGIc:PATtern:DELTatime**

Sets or returns the pattern trigger delta time value. The time value is used as part of the pattern trigger condition to determine if the duration of a logic pattern meets the specified time constraints.

**Group** Trigger
Commands Listed in Alphabetical Order

**Syntax**

TRIGger:A:LOGIc:PATtern:DELTatime <NR3>
TRIGger:A:LOGIc:PATtern:DELTatime?

**Arguments**

<NR3> is a floating point value with exponent that sets the pattern trigger time value. This argument has a range of 39.6E–9 (39.6 ns) to 10.0E0 (10 s), in increments of 13.2 ns. Values that are not an increment of 13.2 ns are rounded to the nearest correct value.

**Examples**

TRIGGER:A:LOGIC:PATTERN:DELTATIME 71.28E-8 sets the pattern trigger delta time value to 712.8 ns.

**TRIGger:A:LOGIc:PATtern:INPut:D<x>**

Sets or returns the A logic trigger input for the specified digital channel <x>, where x is the channel number. This command species the logic value used when the pattern trigger detects the threshold level.

**Group**

Trigger

**Syntax**

TRIGger:A:LOGIC:PATTERN:INPut:D<x> {HIGH|LOW|X}
TRIGger:A:LOGIC:PATTERN:INPut:D<x>?

**Arguments**

HIGH specifies a logic high.
LOW specifies a logic low.
X specifies a “do not care” state.

**TRIGger:A:LOGIc:PATtern:WHEn**

Sets or returns the pattern logic condition on which to trigger the oscilloscope.

**Group**

Trigger

**Syntax**

TRIGger:A:LOGIC:PATTERN:WHEn {TRUE|FALSE|LESSThan|MOREThan|EQUAL|UNEQual}
TRIGger:A:LOGIC:PATTERN:WHEn?

**Arguments**

TRUE triggers the oscilloscope when the pattern becomes true.
FALSE triggers the oscilloscope when the pattern becomes false.
LESSTHAN triggers the oscilloscope when the input pattern is true for a time period less than the time period specified in TRIGGER:A:LOGIC:PATTERN:DELTATIME.

MORETHAN triggers the oscilloscope when the input pattern is true for a time period more (greater) than the time period specified in TRIGGER:A:LOGIC:PATTERN:DELTATIME.

EQUAL triggers the oscilloscope when the input pattern is true for a time period equal to the time period specified in TRIGGER:A:LOGIC:PATTERN:DELTATIME, within a ±5% tolerance.

UNEQUAL triggers the oscilloscope when the input pattern is true for a time period greater than or less than (not equal to) the time period specified in TRIGGER:A:LOGIC:PATTERN:DELTATIME, within a ±5% tolerance.

**Examples**

TRIGGER:A:LOGIC:PATTERN:WHEN LESSTHAN sets the oscilloscope to trigger when the pattern is true for a time period less than the pattern trigger delta time setting.

TRIGger:A:LOGIc:PATtern:WHen:LESSLimit

Sets or returns the maximum time that the selected pattern may be true and still generate an A logic pattern trigger.

**Group**

Trigger

**Syntax**

TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLimit <NR3>
TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLimit?

**Arguments**

<NR3> specifies the maximum amount of time to hold the pattern true.

**Examples**

TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT 10.0E+00 sets the maximum time that the selected pattern may hold true (and generate an A logic pattern trigger) to 10 s.


TRIGger:A:LOGIc:PATtern:WHen:MORELimit

Sets or returns the minimum time that the selected pattern may be true and still generate an A logic pattern trigger.
Commands Listed in Alphabetical Order

Group  Trigger

Syntax  TRIGger:A:LOGIC:PAtrern:WHEn:MORELimit <NR3>
        TRIGger:A:LOGIC:PAtrern:WHEn:MORELimit?

Arguments  <NR3> specifies the minimum amount of time to hold the pattern true.

Examples  TRIGGER:A:LOGIC:PAtrern:WHEn:MORELimit 10.0E+00 sets the minimum time that the selected pattern may hold true (and generate an A logic pattern trigger) to 10s.

        TRIGGER:A:LOGIC:PAtrern:WHEn:MORELimit 8.0000E-9 indicating that the selected pattern must hold true for at least 8 ns to generate an A logic pattern trigger.

TRIGger:A:LOGIC:THReshold:CH<x>

This command sets or queries the trigger A logic threshold voltage for the specified channel x.

Group  Trigger

Syntax  TRIGger:A:LOGIC:THReshold:CH<x> {<NR3>|ECL|TTL}
        TRIGger:A:LOGIC:THReshold:CH<x>?

Arguments  <NR3> specifies the threshold voltage, in volts.

        ECL specifies a preset ECL high level of –1.3V.
        TTL specifies a preset TTL high level of 1.4V.

Examples  TRIGGER:A:LOGIC:THRESHold:CH2 3.0E-3 sets the A logic trigger threshold voltage for Channel 2 to 3 mV.

        TRIGGER:A:LOGIC:THRESHold:CH3? might return
        :TRIGGER:A:LOGIC:THRESHold:CH3 1.2000E+00, indicating that the A logic trigger threshold voltage for Channel 3 is 1.2 V.
**TRIGger:A:LOGIc:THReshold:D<x>**

Sets or returns the trigger A logic threshold level for the specified digital channel <x>. This command affects all trigger types using the digital channel.

**Group**
Trigger

**Syntax**
TRIGger:A:LOGIc:THReshold:D<x> {<NR3>|ECL|TTL}
TRIGger:A:LOGIc:THReshold:D<x>? 

**Related Commands**
TRIGger:A:LEVel:D<x>

**Arguments**
- <NR3> specifies the threshold level in volts.
- ECL specifies a preset ECL high level of –1.3V.
- TTL specifies a preset TTL high level of 1.4V.

**TRIGger:A:LOWerthreshold:CH<x>**

Sets or returns the lower threshold for the channel selected. Each channel can have an independent level. Used in Runt and Slew Rate triggers as the lower threshold. Used for all other trigger types as the single level/threshold.

**Group**
Trigger

**Syntax**
TRIGger:A:LOWerthreshold:CH<x> {ECL|TTL|<NR3>}
TRIGger:A:LOWerthreshold:CH<x>? 

**Related Commands**
TRIGger:A:LEVel:CH<x>

**Arguments**
- ECL specifies a preset ECL high level of –1.3V.
- TTL specifies a preset TTL high level of 1.4V.
- <NR3> is the clock level, in volts.

**Examples**
TRIGGER:A:LOWERTHRESHOLD:CH2 50E-3 sets the lower limit threshold for CH2 of the pulse runt trigger to 50 mV.
TRIGGER:A:LOWERTHRESHOLD:CH2? might return TRIGGER:A:LOWERTHRESHOLD:CH2 1.2000E-01 indicating that the lower limit threshold for CH2 of the pulse runt trigger is set to 120 mV.

**TRIGger:A:LOWerthreshold:D<x>**

Sets the lower threshold for the digital channel selected. Each channel can have an independent level. Used in runt and slew rate as the lower threshold. Used for all other trigger types as the single level/threshold.

**Group**

Trigger

**Syntax**

TRIGger:A:LOWerthreshold:D<x> {<NR3>|ECL|TTL}

TRIGger:A:LOWerthreshold:D<x>? 

**Arguments**

<NR3> specifies the threshold voltage, in volts.

ECL specifies a preset ECL high level of –1.3V.

TTL specifies a preset TTL high level of 1.4V.

**Examples**

TRIGGER:A:LOWERTHRESHOLD:D1 0.5 sets the lower threshold to 0.5 volts.

TRIGGER:A:LOWERTHRESHOLD:D1? might return

TRIGGER:A:LOWERTHRESHOLD:D1 -1.3000 indicating the lower threshold is set to -1.3 volts.

**TRIGger:A:LOWerthreshold{:EXT|:AUX}**

Sets or returns the lower threshold for the Auxiliary Input. Used for the following trigger types: Runt, Slew Rate.

**Group**

Trigger

**Syntax**

TRIGger:A:LOWerthreshold{:EXT|:AUX} {<NR3>|ECL|TTL}

TRIGger:A:LOWerthreshold{:EXT|:AUX}? 

**Arguments**

ECL specifies a preset ECL high level of –1.3V.

TTL specifies a preset TTL high level of 1.4V.

<NR3> specifies the threshold level in volts.
**TRIGger:A:MODE**

Sets or returns the A trigger mode.

**Group**

Trigger

**Syntax**

```
TRIGger:A:MODE {AUTO|NORMal}
TRIGger:A:MODE?
```

**Related Commands**

TRIGger:A:LEVel

**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 specifies that a valid trigger event must occur before a trigger is generated.

TRIGGER:A:MODE? might return :TRIGGER:A:MODE NORMAL indicating that a valid trigger event must occur before a trigger is generated.

**TRIGger:A:PULse? (Query Only)**

Returns the A pulse trigger parameters.

**Group**

Trigger

**Syntax**

```
TRIGger:A:PULse?
```

**Related Commands**

TRIGger:A:EDGE?, TRIGger:A:LOGIc?

**Examples**


**TRIGger:A:PULse:CLAss**

Sets or returns the type of pulse on which to trigger.

**Group**

Trigger
Commands Listed in Alphabetical Order

**Syntax**

TRIGger:A:PULse:CLASS {RUNt|WIDth|TRANsition}
TRIGger:A:PULse:CLASS?

**Related Commands**


**Arguments**

RUNt triggers when a pulse crosses the first preset voltage threshold but does not cross the second preset threshold before recrossing the first.

WIDth triggers when a pulse is found that has the specified polarity and is either inside or outside the specified time limits.

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.

**Examples**

TRIGGER:A:PULSE:CLASS? might return :TRIGGER:A:PULSE:CLASS WIDTH indicating that a pulse was found that is of the specified polarity and width.

**TRIGger:A:PULSEWidth? (Query Only)**

Returns the width parameters for the pulse width trigger.

**Group** Trigger

**Syntax** TRIGger:A:PULSEWidth?


**TRIGger:A:PULSEWidth:POLarity**

Sets or returns the polarity for the width trigger.

**Group** Trigger

**Syntax** TRIGger:A:PULSEWidth:POLarity {NEGative|POSitive}
TRIGger:A: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:PULSEWidth:SOUrce

Sets or returns the source for the pulse-width trigger.

Group
Trigger

Syntax
TRIGger:A:PULSEwidth:SOURce {CH1|CH2|CH3|CH4|LINE|EXT}
TRIGger:A:PULSEWidth:SOURce?

Arguments
CH1–CH4 specifies an analog input channel as the A edge trigger source.
EXT specifies an external trigger using the Aux In connector located on the front panel of the oscilloscope.
LINE specifies AC line voltage.

Examples
TRIGGER:A:PULSEWIDTH:SOURCE CH1 sets channel 1 as the pulse width source.

TRIGger:A:PULSEWidth:WHEn

Sets or returns whether to trigger on a pulse that meets, falls outside, or within the specified range of limits.

Group
Trigger

Syntax
TRIGger:A:PULSEwidth:WHEN {LESSthan|than|EUQal|UNEQual}
TRIGger:A:PULSEWidth:WHEN?

Related Commands
TRIGger:A:PULSEWidth:WIDth
Arguments

- **LESS**\(\text{than}\) argument sets the oscilloscope to trigger if a pulse is detected with width less than the time set by the TRIGger:A:PULSEWidth:WIDth command.
- **than** argument sets the oscilloscope to trigger if a pulse is detected with width than the time set by the TRIGger:A:PULSEWidth:WIDth command.
- **EQUal** argument sets the oscilloscope to trigger if a pulse is detected with width equal to the time period specified in TRIGger:A:PULSEWidth:WIDth within a ±5% tolerance.
- **NOTE**\(\text{qual}\) argument sets the oscilloscope to trigger if a pulse is detected with width greater than or less than (but not equal) the time period specified in TRIGger:A:PULSEWidth:WIDth within a ±5% tolerance.

Examples

- TRIGGER:A:PULSEWIDTH:WHEN LESS\(\text{than}\) specifies that the duration of the A pulse will fall within defined high and low limits.

TRIGger:A:PULSEWidth:WIDth

Sets or returns the width setting for the pulse width trigger.

Group

Trigger

Syntax

TRIGger:A:PULSEWidth:WIDth <NR3>

TRIGger:A:PULSEWidth:WIDth?

Related Commands

TRIGger:A:PULSEWidth:WHEn

Arguments

- <NR3> specifies the pulse width in seconds.

Examples

- TRIGGER:A:PULSEWIDTH:WIDTH 5.0E-6 sets the pulse width to 5 µs.

TRIGger:A:RUNT? (Query Only)

Returns the current A runt trigger parameters.
Groups

**Trigger**

Syntax

TRIGger:A:RUNT?

Examples


TRIGger:A:RUNT:POLarity

Sets or returns the polarity for the runt trigger.

Group

Trigger

Syntax

TRIGger:A:RUNT:POLarity {EITHER|NEGative|POSitive}

TRIGger:A: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:RUNT:SOUrce

Sets or returns the source for the A runt trigger.

Group

Trigger

Syntax

TRIGger:A:RUNT:SOURce {CH1|CH2|CH3|CH4}

TRIGger:A:RUNT:SOURce?
**Arguments**

CH1–CH4 specifies the input channel number, depending on the model of the oscilloscope.

**Examples**

TRIGGER:A:RUNT:SOURCE CH4 sets channel 4 as the source for the A pulse trigger.


**TRIGger:A:RUNT:WHEn**

Sets or returns the type of pulse width the trigger checks for when it detects a runt.

**Group**

Trigger

**Syntax**

TRIGger:A:RUNT:WHEn {LESSthan|than|EQual|UNEQual|OCCURS}

TRIGger:A:RUNT:WHEn?

**Related Commands**

TRIGger:A: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:RUNT:WIDth command.

than argument sets the oscilloscope to trigger if the a runt pulse is detected with width than the time set by the TRIGger:A:RUNT:WIDth command.

EQUa1 argument sets the oscilloscope to trigger if a runt pulse is detected with width equal to the time period specified in TRIGger:A:RUNT:WIDth within a ±5% tolerance.

NOTEqua1 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:RUNT:WIDth within a ±5% tolerance.

**Examples**

TRIGGER:A:RUNT:WHEN THAN sets the runt trigger to occur when the oscilloscope detects a runt in a pulse wider than the specified width.

**TRIGger:A:RUNT:WIDth**

Sets or returns the width for a runt trigger.

**Group**
Trigger

**Syntax**
TRIGger:A:RUNT:WIDth <NR3>
TRIGger:A:RUNT:WIDth?

**Related Commands**
TRIGger:A:RUNT:WHEn

**Arguments**
<NR3> 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:SETHold? (Query Only)**

Returns the clock edge polarity, voltage threshold and source input; data voltage threshold and source; and both setup and hold times for setup and hold violation triggering.

**Group**
Trigger

**Syntax**
TRIGger:A:SETHold?

**Examples**
TRIGGER:A:SETHOLD? might return
:TRIGGER:A:SETHOLD:CLOCK:SOURCE CH1;EDGE RISE;THRESHOLD 100.0000E-3
:TRIGGER:A:SETHOLD:DATA:SOURCE CH2;THRESHOLD 80.0000E-3;
:TRIGGER:A:SETHOLD:HOLDTIME 20.0000E-9;SETHOLD 8.0000E-9

**TRIGger:A:SETHold:CLOCk? (Query Only)**

Returns the clock edge polarity, voltage threshold, and source input for setup and hold triggering.
Commands Listed in Alphabetical Order

<table>
<thead>
<tr>
<th>Group</th>
<th>Trigger</th>
</tr>
</thead>
</table>

**Syntax**

```
TRIGger:A:SETHold:CLOCK?
```

**Examples**

```
TRIGGER:A:SET HOLD:CLOCK? might return
:TRIGGER:A:SET HOLD:CLOCK:SOURCE EXT;EDGE FALL;THRESHOLD 1.4000
```

---

### TRIGger:A:SETHold:CLOCK:EDGE

Sets or returns the clock edge polarity for setup and hold triggering.

<table>
<thead>
<tr>
<th>Group</th>
<th>Trigger</th>
</tr>
</thead>
</table>

**Syntax**

```
TRIGger:A:SETHold:CLOCK:EDGE {FALL|RISe}
TRIGger:A: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:SOURce

Sets or returns the clock source for the setup and hold triggering.

<table>
<thead>
<tr>
<th>Group</th>
<th>Trigger</th>
</tr>
</thead>
</table>

**Syntax**

```
TRIGger:A:SETHold:CLOCK:SOURce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15|
TRIGger:A:SETHold:CLOCK:SOURce?
```

**Related Commands**

```
TRIGger:A:SETHold:DATa:SOURce
```
### TRIGger:A:SETHold:CLOCK:THReshold

Sets or returns the clock voltage threshold for the setup and hold trigger.

#### Group
Trigger

#### Syntax
```
TRIGger:A:SETHold:CLOCK:THReshold {<NR3>|TTL}
TRIGger:A:SETHold:CLOCK:THReshold?
```

#### Arguments
- **TTL** specifies a preset TTL high level of 1.4 V.
- **<NR3>** is the clock level, in volts.

#### Examples
- `TRIGGER:A:SETHOLD:CLOCK:THRESHOLD TTL` specifies the preset TTL value of 1.4 V as the clock threshold for the setup and hold trigger.

### TRIGger:A:SETHold:DATa? (Query Only)

Returns the voltage threshold and data source for the setup and hold trigger.

#### Group
Trigger

#### Syntax
```
TRIGger:A:SETHold:DATa?
```
Commands Listed in Alphabetical Order

**Related Commands**

TRIGger:A:SETHold:CLOCk?

**Examples**

TRIGGER:A:SETHOLD:DATA? might return
:TRIGGER:A:SETHOLD:DATA:SOURCE CH2;THRESHOLD 80.0000E-3

**TRIGger:A:SETHold:DATa:SOUrce**

Sets or returns the data source for the setup and hold trigger. You cannot specify the same source for both clock and data.

For DPO models, you can specify only a single data source. Data sources for DPO models may be one of CH1-CH4 or the Auxin port (EXT or AUX).

For MSO models, you can specify any combination of CH1-CH4 and D0-D15 as the data sources.

**Group**

Trigger

**Syntax**

DPO Models:
TRIGger:A:SETHold:DATa:SOUrce <wfm>

MSO Models:
TRIGger:A:SETHold:DATa:SOUrce <wfm>[,<wfm>][,<wfm>][,<wfm>][,<wfm>][,<wfm>][,<wfm>]

TRIGger:A:SETHold:DATa:SOUrce?

**Related Commands**

TRIGger:A:SETHold:CLOCk:SOUrce

**Arguments**

DPO Models:
<wfm> specifies the source channel number and is one of CH1-CH4, EXT or AUX. You can specify only one waveform on a DPO.

MSO Models:
<wfm> specifies the source channel numbers. Each <wfm> can be one of CH1-CH4 or D0-D15. Auxin (EXT or AUX) is not valid as a data source.

**Examples**

TRIGGER:A:SETHOLD:DATA:SOURCE CH1 sets channel 1 as the data source for the setup and hold trigger.

:TRIGGER:A:LOGIC:SETHOLD:DATA:SOURCE CH2 indicating that channel 2 is the current data source for the setup and hold trigger.
**TRIGger:A:SETHold:DATa:THReshold**

Sets or returns the data voltage threshold for setup and hold trigger.

**Group**  
Trigger

**Syntax**  
TRIGger:A:SETHold:DATa:THReshold {<NR3>|TTL}  
TRIGger:A:SETHold:DATa:THReshold?

**Arguments**  
TTL specifies the preset TTL high level of 1.4 V.  
<NR3> is the setup and hold data level, in V.

**Examples**  
TRIGGER:A:SETHOLD:DATA:THRESHOLD TTL specifies the preset high level of 1.4 V as the current data voltage level for the setup and hold trigger.  
:TRIGGER:A:SETHOLD:DATA:THRESHOLD 1.2000E+00 indicating that 1.2 V is the current data voltage level for the setup and hold trigger.

**TRIGger:A:SETHold:HOLDTime**

Sets or returns the hold time for setup and hold violation triggering.

**Group**  
Trigger

**Syntax**  
TRIGger:A:SETHold:HOLDTime <NR3>  
TRIGger:A:SETHold:HOLDTime?

**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**  
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:SETHold:SETTime**

Sets or returns the setup time for setup and hold violation triggering.

**Group** Trigger

**Syntax**

```
TRIGger:A:SETHold:SETTime <NR3>
TRIGger:A:SETHold:SETTime?
```

**Arguments**

<NR3> 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:SETHOLD:SETTIME?
```

might return

```
:TRIGGER:A:LOGIC:SETHOLD:SETTIME 2.0000E-09
```

indicating that the current setup time for setup and hold trigger is 2 ns.

**TRIGger:A:SETHold:THReshold:CH<x>**

Sets or queries the threshold for the channel specified by x. Affects all trigger types using the channel.

**Group** Trigger

**Syntax**

```
TRIGger:A:SETHold:THReshold:CH<x> {<NR3>|ECL|TTL}
TRIGger:A:SETHold:THReshold:CH<x>? 
```

**Arguments**

<NR3> specifies the threshold voltage, in volts.

ECL specifies a preset ECL high level of –1.3V.

TTL specifies a preset TTL high level of 1.4V.

**Examples**

```
TRIGGER:A:SETHOLD:THRESHOLD:CH1 1.5
```

sets the channel 1 threshold to 1.5 volts.

```
TRIGGER:A:SETHOLD:THRESHOLD:CH1?
```

might return

```
TRIGGER:A:SETHOLD:THRESHOLD:CH1 0.0E+0
```

indicating the channel 1 threshold is set to 0.0 volts.
**TRIGger:A:SETHold:THReshold:D<x>**

Sets the threshold for the selected digital channel. Affects all trigger types using the digital channel.

**Group**  
Trigger

**Syntax**  
TRIGger:A:SETHold:THReshold:D<x> {<NR3>|ECL|TTL}  
TRIGger:A:SETHold:THReshold:D<x>?

**Arguments**  
<x> specifies the threshold voltage, in volts.

- ECL specifies a preset ECL high level of –1.3V.
- TTL specifies a preset TTL high level of 1.4V.

**Examples**  
TRIGGER:A:SETHOLD:THRESHOLD:D1 ECL sets the threshold to ECL levels.

TRIGGER:A:SETHOLD:THRESHOLD:D1? might return

TRIGGER:A:SETHOLD:THRESHOLD:D1 -1.3000 indicating the threshold is set to -1.3 volts.

**TRIGger:A{:TRANSition|:RISEFall}? (Query Only)**

Returns transition time trigger parameters.

**Group**  
Trigger

**Syntax**  
TRIGger:A{:TRANSition|:RISEFall}?  

**Related Commands**  
TRIGger:A:UPPerthreshold:CH<x> , TRIGger:A:LOWerthreshold:CH<x>

**Examples**  
TRIGGER:A::TRANSITION? might return

:TRIGGER:A:TRANSITION:POLARITY POSITIVE;WHEN SLOWER;DELTATIME 8.0000E-9

**TRIGger:A{:TRANSition|:RISEFall}:DELTatime**

Sets or returns the delta time used in calculating the transition value for the transition trigger.
Commands Listed in Alphabetical Order

**Group** Trigger

**Syntax**

TRIGger:A{:TRANSition|:RISEFall}:DELTatime <NR3>

TRIGger:A{:TRANSition|:RISEFall}:DELTatime?

**Arguments**

<NR3> 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{:TRANSition|:RISEFall}:POLarity**

Sets or returns the polarity for the transition trigger.

**Group** Trigger

**Syntax**

TRIGger:A{:TRANSition|:RISEFall}:POLarity {EITher|NEGative|POSitive}

TRIGger:A{:TRANSition|:RISEFall}: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{:TRANSition|:RISEFall}:SOUrce**

Sets or returns the source for transition trigger.
**Group**
Trigger

**Syntax**
TRIGger:A{:TRANSition|:RISEFall}:SOUrce \{CH1|CH2|CH3|CH4\}
TRIGger:A{:TRANSition|:RISEFall}:SOUrce?

**Arguments**
CH1–CH4 specifies one of the input channels.

**Examples**
TRIGGER:A:TRANSITION:SOURCE CH4 sets channel 4 as the source for the transition trigger.


---

**TRIGger:A{:TRANSition|:RISEFall}:WHEn**
Sets or returns whether to check for a transitioning signal that is faster or slower than the specified delta time.

**Group**
Trigger

**Syntax**
TRIGger:A{:TRANSition|:RISEFall}:WHEn \{SLOWer|FASTer|EQual|UNEQual\}
TRIGger:A{:TRANSition|:RISEFall}:WHEn?

**Arguments**

- **FASTER** sets the trigger to occur when the signal transition time is faster than the time set by TRIGger:A{:TRANSition|:RISEFall}:DELTatime.

- **SLOWER** sets the trigger to occur when the signal transition time is slower than the time set by TRIGger:A{:TRANSition|:RISEFall}:DELTatime.

- **EQUAL** sets the trigger to occur when the signal transition time is equal to the time set by TRIGger:A{:TRANSition|:RISEFall}:DELTatime.

- **UNEQUAL** sets the trigger to occur when the signal transition time is not equal to the time set by TRIGger:A{:TRANSition|:RISEFall}: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|:RISEFall}:DELTatime command.

**TRIGger:A:TYPe**

Sets or returns the type of A trigger. Once you have set the trigger type, you may also need to identify the associated trigger class. For details on selecting Logic and Pulse trigger classes, see TRIGger:A:LOGIc:CLAss and TRIGger:A:PULse:CLAss respectively.

**Group** Trigger

**Syntax**

```
TRIGger:A:TYPe {EDGe|LOGic|PULSe|BUS|VIDeo}
TRIGger:A:TYPe?
```

**Related Commands**


**Arguments**

**EDGe** is the default 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.

**LOGic** specifies that a trigger occurs when specified conditions are met and is controlled by the TRIGger:A:LOGIc? commands. This trigger type is equivalent to the logic trigger and to the setup and hold trigger found in the user interface.

**PULSe** specifies that a trigger occurs when a specified pulse is found and is controlled by the TRIGger:A:PULse? commands. This trigger type is equivalent to the pulse width, runt, and rise/fall time triggers found in the user interface.

**BUS** specifies that a trigger occurs when a communications signal is found. Supports CAN, I2C, SPI, and RS-232 communications signals.

**VIDeo** specifies that the trigger occurs when a video signal is found.

**Examples**

```
TRIGGER:A:TYPe EDGE sets the A trigger type to EDGE.

TRIGGER:A:TYPe? might return :TRIGGER:A:TYPe PULSE indicating that the A trigger type is a pulse trigger.
```

**TRIGger:A:UPPerthreshold:CH<x>**

Sets the upper threshold for channel <x>, where x is the channel number. Each channel can have an independent level. Used only for runt and slew rate trigger types.

**Group** Trigger
Syntax

```
TRIGGER:A:UPPERthreshold:CH<x> {<NR3>|ECL|TTL}
TRIGGER:A:UPPERthreshold:CH<x>?
```

Arguments

- `<NR3>` is the threshold level in volts.
- ECL specifies a preset ECL high level of –1.3V.
- TTL specifies a preset TTL high level of 1.4V.

Examples

```
TRIGGER:A:UPPERthreshold:CH2 50E-3 sets the upper limit of the pulse runt trigger to 50 mV for channel 2.
TRIGGER:A:UPPERthreshold:CH2? might return :TRIGGER:A:UPPERthreshold:CH2 1.2000E-01 indicating that the upper limit of the pulse runt trigger is set to 120 mV.
```

**TRIGGER:A:VIDeo? (Query Only)**

Returns the A trigger video parameters.

Group

Trigger

Syntax

```
TRIGGER:A:VIDeo?
```

Examples

```
```

**TRIGGER:A:VIDeo:CUStom{:FORMat|:TYPE}**

Sets or returns the video trigger format. Use this command only when the video format is set to custom.

Conditions

This command requires a DPO4VID application module.

Group

Trigger

Syntax

```
TRIGGER:A:VIDeo:CUStom{:FORMat|:TYPE} {INTERLAced|PROGressive}
```
TRIGger:A:VIDeo:CUSTom{:FORMAT|:TYPE}?  

Arguments  
INTERLaced argument sets the format for interlaced video lines.  
PROgressive argument sets the format for progressive video lines.  

Examples  
TRIGGER:A:VIDEO:CUSTOM:FORMAT PROGRESSIVE sets the custom format for the A video trigger to progressive lines.  

TRIGger:A:VIDeo:CUSTom:LINEPeriod  
Sets or queries the line period for the standard under test. Use this command only when the video format is set to custom.  

Conditions  
This command requires a DPO4VID application module.  

Group  
Trigger  

Syntax  
TRIGger:A:VIDeo:CUSTom:LINEPeriod <NR3>  
TRIGger:A:VIDeo:CUSTom:LINEPeriod?  

Arguments  
<NR3> is the custom video line period.  

Examples  
TRIGGER:A:VIDEO:CUSTOM:LINEPERIOD 50.5E-6 sets the video line period to 50.5 μs.  

TRIGger:A:VIDeo:CUSTom:SCAN  
Sets or returns the horizontal line scan rate of the A video trigger. Use this command only when the video format is set to custom.  

Conditions  
This command requires a DPO4VID application module.
Commands Listed in Alphabetical Order

**Group**  
Trigger

**Syntax**  
`TRIGger:A:VIDeo:CUSTom:SCAN {RATE15K|RATE20K|RATE25K|RATE35K|RATE50K}`

`TRIGger:A:VIDeo:CUSTom:SCAN?`

**Arguments**  
**RATE15** sets the range of the video line scan rate to 15 kHz through 20 kHz. This is the standard broadcast rate.

**RATE20** sets the range of the video line scan rate to 20 kHz through 25 kHz.

**RATE25** sets the range of the video line scan rate to 25 kHz through 35 kHz.

**RATE35** sets the range of the video line scan rate to 35 kHz through 50 kHz.

**RATE50** sets the range of the video line scan rate to 50 kHz through 65 kHz.

**Examples**  
`TRIGGER:A:VIDEO:CUSTOM:SCAN RATE15` sets the scan rate of the A trigger custom video to Rate 1, which is 15 kHz to 20 kHz (standard broadcast rate).

`TRIGGER:A:VIDEO:CUSTOM:SCAN?` might return `:TRIGGER:A:VIDEO:CUSTOM:SCAN RATE20` indicating that the video line rate for the A trigger custom video is set to Rate20, which is 20 kHz to 25 kHz.

**TRIGger:A:VIDeo:CUSTom:SYNCInterval**

Sets or queries the sync interval for the standard under test. This is only required for BiLevel Custom. Use this command only when the video format is set to custom.

**Conditions**  
This command requires a DPO4VID application module.

**Group**  
Trigger

**Syntax**  
`TRIGger:A:VIDeo:CUSTom:SYNCInterval <NR3>`

`TRIGger:A:VIDeo:CUSTom:SYNCInterval?`

**Arguments**  
**<NR3>** is the sync interval.

**Examples**  
`TRIGGER:A:VIDEO:CUSTOM:SYNCINTERVAL 4.0E-6` sets the sync interval is set to 4.0 μs.
TRIGGER:A:VIDEO:CUSTOM:SYNCINTERVAL 4.7200E–6 indicating the sync interval is set to 4.72 μs.

**TRIGger:A:VIDeo:HDtv:FORMat**

Sets or returns the HDTV video signal format on which to trigger.

**Conditions**
This command requires a DPO4VID application module.

**Group**
Trigger

**Syntax**
TRIGger:A:VIDeo:HDtv:FORMat {HD1080P24|HD720P60|HD480P60|HD1080I50|HD1080P25| HD1080I60|HD1080PSF24}

Arguments

<table>
<thead>
<tr>
<th>HDTV format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1080i50</td>
<td>1125 Lines (1080 active), 1920 x 1080 pixel, interlaced, 60 fps</td>
</tr>
<tr>
<td>1080i60</td>
<td>1125 lines (1080 active), 1920 x 1080 pixel, interlaced, 50 fps</td>
</tr>
<tr>
<td>1080p24</td>
<td>1125 lines (1080 active), 1920 x 1080 pixel, progressive, 24 fps</td>
</tr>
<tr>
<td>1080p25</td>
<td>1125 lines (1080 active), 1920 x 1080 pixel, progressive, 25 fps</td>
</tr>
<tr>
<td>1080sf24</td>
<td>1125 Lines (1080 active), 1920 x 1080 pixel, progressive (sF), 24 fps</td>
</tr>
<tr>
<td>720p60</td>
<td>750 lines (720 active), 1280 x 720 pixel, progressive, 60 fps</td>
</tr>
<tr>
<td>480p60</td>
<td>525 lines (480 active), 640 or 704 x 480 pixel, progressive, 60 fps</td>
</tr>
</tbody>
</table>

**TRIGger:A:VIDeo:HOLDoff:FIELD**

Sets or returns the video trigger holdoff in terms of video fields.

**Conditions**
This command requires a DPO4VID application module.
Commands Listed in Alphabetical Order

<table>
<thead>
<tr>
<th>Group</th>
<th>Trigger</th>
</tr>
</thead>
</table>
| Syntax        | TRIGger:A:VIDeo:HOLDoff:FIELD <NR3>  
|               | TRIGger:A:VIDeo:HOLDoff:FIELD? |
| Arguments     | <NR3> argument is a real number from 0.0 to 8.5 in increments of 0.5. The argument sets the number of fields that the oscilloscope waits before rearming the video trigger. |
|               | :TRIGger:A:VIDeo:HOLdoff:FIELD 5 indicating that the oscilloscope is set to wait 5 video fields before rearming the trigger.  
|               | TRIGGER:A:VIDEO:HOLDOFF:FIELD 4.5 sets the oscilloscope to wait 4.5 video fields before rearming the trigger. |

**TRIGger:A:VIDeo:LINE**

Sets or returns the video line number on which the oscilloscope triggers. Use the TRIGger:A:VIDeo{:SYNC|:FIELD} command to actually trigger the oscilloscope on the line that you specify with this command.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>This command requires a DPO4VID application module.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Trigger</td>
</tr>
</tbody>
</table>
| Syntax        | TRIGger:A:VIDeo:LINE <NR1>  
|               | TRIGger:A:VIDeo:LINE? |
| Related Commands | TRIGger:A:VIDeo{:SYNC|:FIELD} |
| Arguments     | <NR1> argument is an integer that sets the video line number on which the oscilloscope triggers. The following table lists the valid choices, depending on the active video standard. |

**Table 2-44: Video Line Numbering Ranges**

<table>
<thead>
<tr>
<th>Video Standard</th>
<th>Line Number Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>525/NTSC</td>
<td>1–525</td>
</tr>
<tr>
<td>625/PAL, SECAM</td>
<td>1–625</td>
</tr>
<tr>
<td>SECAM</td>
<td>1–625</td>
</tr>
</tbody>
</table>
**Examples**

TRIGGER:A:VIDEO:LINE 23 sets the oscilloscope to trigger on the line 23.


**TRIGger:A:VIDeo:POLarity**

Sets or returns the polarity of the A video trigger.

**Conditions**

This command requires a DPO4VID application module.

**Group**

Trigger

**Syntax**

TRIGger:A:VIDeo:POLarity {NEGative|POSitive}

TRIGger:A:VIDeo:POLarity?

**Arguments**

POSitive argument sets the oscilloscope to trigger on a positive video sync pulse.

NEGative argument sets the oscilloscope to trigger on a negative video sync pulse.

**Examples**

TRIGGER:A:VIDEO:POLARITY NEGATIVE sets the oscilloscope to trigger on a negative video pulse.

TRIGGER:A:VIDEO:POLARITY? might return :TRIGger:A:VIDeo:POLARITY POSITIVE indicating that the oscilloscope is set to trigger on a positive video sync pulse.

**TRIGger:A:VIDeo:SOUrce**

Sets or returns the source for the A video trigger.

**Group**

Trigger

**Syntax**

TRIGger:A:VIDeo:SOUrce {CH1|CH2|CH3|CH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}

TRIGger:A:VIDeo:SOUrce?

**Arguments**

CH1–CH4 or D0–D15 specifies the input channel to use as the A video trigger.
Examples

TRIGGER:A:VIDEO:SOURCE CH1 sets the source for A video trigger to Channel 1.


TRIGger:A:VIDeo:STANdard

Sets or returns the standard for the video trigger.

Group

Trigger

Syntax

TRIGger:A:VIDeo:STANdard {NTSc|PAL|SECAM|CUSTom|HDtv}

Arguments

NTSc sets the oscilloscope to trigger on video signals that meet the NTSC 525/60/2:1 standard (a line rate of 525 lines per frame and a field rate of 60 Hz).

PAL sets the oscilloscope to trigger on video signals that meet the NTSC 625/50/2:1 standard (a line rate of 625 lines per frame and a field rate of 50 Hz).

SECAM sets the oscilloscope to trigger on video signals that meet the SECAM standard.

CUSTom sets the oscilloscope to trigger on video horizontal scan rate parameters defined by TRIGger:A:VIDeo:CUSTom:SCAN command.

HDtv sets the oscilloscope to trigger on HDTV video signals that meet standards defined by the TRIGger:A:VIDeo:HDtv:FORMat command.

Examples

TRIGGER:A:VIDEO:STANDARD NTSC sets the oscilloscope to trigger on NTSC-standard video signals.


TRIGger:A:VIDeo{:SYNC|:FIELD}

Sets or returns the video field or line that the trigger detects.

Group

Trigger

Syntax

TRIGger:A:VIDeo{:SYNC|:FIELD} {ODD|EVEN|ALLFields|ALLLines|NUMERic}

TRIGger:A:VIDeo{:SYNC|:FIELD}?
Arguments

**ODD** argument sets the oscilloscope to trigger on interlaced video odd fields.

**EVEN** argument sets the oscilloscope to trigger on interlaced video even fields.

**ALLFields** argument sets the oscilloscope to trigger on all fields.

**ALLLines** argument sets the oscilloscope to trigger on all video lines.

**NUMERic** argument sets the oscilloscope to trigger on the video signal line specified by the **TRIGger:A:VIDeo:LINE** command.

Examples

**TRIGGER:A:VIDEO:FIELD** **EVEN** sets the A video trigger so that it will trigger on even fields.

**TRIGGER:A:VIDEO:FIELD?** might return **:TRIGGER:A:VIDEO:FIELD ALLFIELDS** indicating that the A video will trigger on all video fields.

**TRIGger:B**

Sets the B trigger level to 50% of minimum and maximum. The query form of this command returns the B trigger parameters. This command is similar to selecting B Event (Delayed) Trigger Setup from the Trig menu and then viewing the current setups.

Related Commands **TRIGger:A**

Arguments

**SETLevel** sets the B trigger level to 50% of MIN and MAX.

Examples

**TRIGGER:B SETLEVEL** sets the B trigger level to 50% of MIN and MAX.

**TRIGGER:B?** might return the following B trigger parameters:

:TRIGGER:B:STATE 0;TYPE EDGE; LEVEL -220.0000E-3;BY TIME;EDGE:SOURCE CH1;SLOPE RISE;COUPLING DC; :TRIGGER:B:TIME 16.0000E-9;EVENTS:COUNT 2

**TRIGger:B:BY**

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

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:EDGE? (Query Only)**

Returns the source, slope, and coupling for B trigger.

Group: Trigger

Syntax: TRIGger:B:EDGE?

Related Commands: TRIGger:B:EDGE:COUPling, TRIGger:B:EDGE:SLOpe, TRIGger:B:EDGE:SOUrce

Examples: TRIGGER:B:EDGE? might return :TRIGGER:B:EDGE:SOURCE CH1; SLOPE RISE; COUPLING DC

**TRIGger:B:EDGE:COUPling**

Sets or returns the type of coupling for the B trigger.

Group: Trigger
Syntax

```
TRIGger:B:EDGE:COUPling {DC|HFRej|LFRej|NOISErej}
TRIGger:B:EDGE:COUPling?
```

Related Commands

```
TRIGger:B:EDGE?
```

Arguments

- **DC** selects DC trigger coupling.
- **HFRej** selects high-frequency reject coupling.
- **LFRej** selects low-frequency reject coupling.
- **NOISErej** selects DC low sensitivity.

Examples

```
TRIGGER:B:EDGE:COUPLING DC selects DC for the B trigger coupling.
```

**TRIGger:B:EDGE:SLOpe**

Sets or returns the slope for the B trigger.

Group

Trigger

Syntax

```
TRIGger:B:EDGE:SLOpe {RISe|FALL}
TRIGger:B:EDGE:SLOpe?
```

Related Commands

```
TRIGger:B:EDGE?
```

Arguments

- **RISe** triggers on the rising or positive edge of a signal.
- **FALL** triggers on the falling or negative edge of a signal.

Examples

```
TRIGGER:B:EDGE:SLOPE FALL sets the B edge trigger to occur on the falling slope.
TRIGGER:B:EDGE:SLOPE? might return :TRIGGER:B:EDGE:SLOPE RISE indicating that the B edge trigger occurs on the rising slope.
```

**TRIGger:B:EDGE:SOUrce**

Sets or returns the source for the B trigger.
Group        Trigger

Syntax       TRIGger:B:EDGE:SOURce {CH<x>|EXT|LINE}
             TRIGger:B:EDGE:SOURce?

Related Commands       TRIGger:B:EDGE?

Arguments       CH<x> specifies one of the input channels as the B trigger source.
                 EXT specifies an external trigger (using the Aux In connector, located on the front panel of the oscilloscope) as the B trigger source.
                 LINE specifies the power line as the B trigger source.

Examples       TRIGGER:B:EDGE:SOURce CH4 sets channel 4 as the input source for the B trigger.
                 TRIGGER:B:EDGE:SOURce? might return :TRIGGER:B:EDGE:SOURce CH1 indicating that the current input source for the B trigger is channel 1.

TRIGger:B:EVENTS? (Query Only)

Returns the current B trigger events parameter.

Group        Trigger

Syntax       TRIGger:B:EVENTS?

Related Commands       TRIGger:B:EVENTS:COUNt

Examples       TRIGGER:B:EVENTS? might return :TRIGGER:B:EVENTS:COUNt 2 indicating that 2 events must occur before the B trigger occurs.

TRIGger:B:EVENTS:COUNt

Sets or returns the number of events that must occur before the B trigger (when TRIG:DELay:BY is set to EVENTS).
**TRIGger:B:EVENTS:COUNt**

Sets or returns the number of B trigger events.

**Group** Trigger

**Syntax**

- TRIGger:B:EVENTS:COUNt <NR1>
- TRIGger:B:EVENTS:COUNt?

**Related Commands** TRIGger:B:EVENTS?

**Arguments**

<NR1> is the number of B trigger events, which can range from 1 to 10,000,000.

**Examples**

TRIGGER:B:EVENTS:COUNT 4 specifies that the B trigger will occur four trigger events after the A trigger.

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:LEVEL**

Sets or returns the level for the B trigger.

**Group** Trigger

**Syntax**

- TRIGger:B:LEVEL {TTL|<NR3>}
- TRIGger:B:LEVEL?

**Related Commands** TRIGger:A:LEVEL, TRIGger:B, TRIGger:B:EDGE:SOURce

**Arguments**

TTL specifies a preset TTL high level of 1.4 V.

<NR3> is the B trigger level, in volts.

**Examples**

TRIGGER:B:LEVEL TTL sets the B trigger level to 1.4 V.

TRIGGER:B:LEVEL? might return :TRIGGER:B:LEVEL 173.0000E-03 indicating that the B trigger level is currently set at 173 mV.

**TRIGger:B:LEVEL:CH<x>**

Sets or returns the B trigger level for channel <x>, where x is the channel number. Each Channel can have an independent Level.
Commands Listed in Alphabetical Order

**Group** Trigger

**Syntax**

```
TRIGger:B:LEVel:CH<x> {ECL|TTL|<NR3>}
TRIGger:B:LEVel:CH<x>?
```

**Arguments**

ECL specifies a preset ECL high level of –1.3V.
TTL specifies a preset TTL high level of 1.4V.
<NR3> specifies the trigger level in user units (usually volts).

**Examples**

```
TRIGGER:B:LEVEL:CH2? might return :TRIGGER:B:LEVEL:CH2 1.3000E+00 indicating that the B edge trigger is set to 1.3 V for channel 2.
TRIGGER:B:LEVEL:CH3 TTL sets the B edge trigger to TTL high level for channel 3.
```

**TRIGger:B:LEVel:D<x>**

Sets or returns the B trigger level for digital channel <x>, where x is the channel number. Each channel can have an independent level.

**Group** Trigger

**Syntax**

```
TRIGger:B:LEVel:D<x> {ECL|TTL|<NR3>}
TRIGger:B:LEVel:D<x>?
```

**Arguments**

ECL specifies a preset ECL high level of –1.3V.
TTL specifies a preset TTL high level of 1.4V.
<NR3> specifies the trigger level in user units (usually volts).

**TRIGger:B:LOWerthreshold:CH<x>**

Sets or returns the B trigger lower threshold for the channel <x>, where x is the channel number. Each channel can have an independent level. Used in runt and slew rate triggers as the lower threshold. Used for all other trigger types as the single level/threshold.

**Group** Trigger
**Syntax**

```
TRIGger:B:LOWerthreshold:CH<x> {ECL|TTL|<NR3>}
TRIGger:B:LOWerthreshold:CH<x>?
```

**Arguments**

- **ECL** specifies a preset ECL high level of –1.3V.
- **TTL** specifies a preset TTL high level of 1.4V.
- **<NR3>** is the threshold level, in volts.

**TRIGger:B:LOWerthreshold:D<x>**

Sets or queries the lower threshold for the digital channel selected. Each channel can have an independent level. Used in runt and slew rate as the lower threshold. Used for all other trigger types as the single level/threshold.

**Group**

Trigger

**Syntax**

```
TRIGger:B:LOWerthreshold:D<x> {<NR3>|ECL|TTL}
TRIGger:B:LOWerthreshold:D<x>?
```

**Arguments**

- **<NR3>** specifies the threshold in volts.
- **ECL** specifies a preset ECL high level of -1.3V.
- **TTL** specifies a preset TTL high level of 1.4V.

**Examples**

```
TRIGGER:B:LOWERTHRESHOLD:D1?  might return
TRIGGER:B:LOWERTHRESHOLD:D1  1.4000 indicating the threshold is set to 1.4 volts.
```

**TRIGger:B:STATE**

Sets or returns 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.

**Group**

Trigger

**Syntax**

```
TRIGger:B:STATE {ON|OFF|<NR1>}
TRIGger:B:STATE?
```

**Related Commands**

- **TRIGger:A:MODe**
**Arguments**

ON specifies that the B trigger is active and in causes trigger events conjunction with the A trigger.

OFF specifies that only the A trigger causes trigger events.

<NR1> a 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**

Sets or returns B trigger delay time. 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

**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:B:TYPe**

Sets or returns the type of B trigger. The only supported B trigger type is EDGE.

**Group**

Trigger

**Syntax**

TRIGger:B:TYPe EDGE

TRIGger:B:TYPe?
Related Commands  TRIGger:A:TYPO

Arguments  EDGE sets the B trigger type to edge.

Examples  TRIGGER:B:TYPE EDGE sets the B trigger type to edge.
           TRIGGER:B:TYPE? might return :TRIGGER:B:TYPE EDGE.

TRIGger:B:UPPerthreshold:CH<x>

Sets the upper threshold for the channel selected. Each channel can have an independent level.

Group  Trigger

Syntax  TRIGger:B:UPPerthreshold:CH<x> {<NR3>|TTL}
        TRIGger:B:UPPerthreshold:CH<x>?

Arguments  TTL specifies a preset TTL high level of 1.4 V.
           <NR3> is the clock level, in volts.

TRIGger:EXTernal? (Query Only)

Returns all external trigger parameters.

Group  Trigger

Syntax  TRIGger:EXTernal?

TRIGger:EXTernal:PRObe

Sets or returns the attenuation factor value of the external probe connector.

Group  Trigger

Syntax  TRIGger:EXTernal:PRObe <NR3>
        TRIGger:EXTernal:PRObe?
Arguments  

<NR3> is the attenuation factor of the probe.

Examples  


TRIGger:EXTer nal:YUNIts? (Query Only)

Returns the external trigger vertical (Y) units value.

Group  

Trigger

Syntax  

TRIGger:EXTer nal:YUNIts?

Examples  

TRIGGER:EXTERNAL:YUNITS? might return TRIGGER:EXTERNAL:YUNITS “V” if the vertical unit is volts.

TRIGger:FREQuency? (Query Only)

Returns the trigger frequency in hertz if available. If the trigger frequency is not currently available, the IEEE Not A Number (NaN = 99.10E+36) value is returned. The maximum precision of the returned frequency is 12 digits.

Group  

Trigger

Syntax  

TRIGger:FREQuency?

Examples  

TRIGGER:FREQUENCY? might return TRIGGER:FREQUENCY 99.10+36 indicating that the trigger frequency is not available.

TRIGger:STATE? (Query Only)

Returns the current state of the triggering system.

Group  

Trigger

Syntax  

TRIGger:STATE?
**Commands Listed in Alphabetical Order**

**Related Commands**

TRIGger:A:MODe

**Returns**

ARMED indicates that the oscilloscope is acquiring pretrigger information.

AUTO indicates that the oscilloscope is in the automatic mode and acquires data even in the absence of a trigger.

READY indicates that all pretrigger information has been acquired and that the oscilloscope is ready to accept a trigger.

SAVE indicates that the oscilloscope is in save mode and is not acquiring data.

TRIGGER indicates that the oscilloscope 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)**

Tests (self-test) the interface and returns a 0.

**Group**

Miscellaneous

**Syntax**

*TST?

**Examples**

*TST? always returns 0.

**UNLock (No Query Form)**

Unlocks the front panel. The command is equivalent to LOCk NONe.

**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.

**USBD:**

**USBDemo:**

Enables or disables the rear USB port for use with Pictbridge printers.

**Group**

PictBridge

**Syntax**

USBDemo: CONFigure {DISabled|IMAge|USBTmc}  
USBDemo: CONFigure?

**Arguments**

DISabled disables the rear USB port.  
IMAge enables the rear USB port as an SIC device.  
USBTmc enables the rear USB port as a USBTMC device.

**Examples**

USBDemo: CONFigure IMAge enables the rear USB port as an SIC device  
USBDemo: CONFigure ? might return USBDemo: CONFigure USBT indicating a USBTMC device.

**USBTMC? (Query Only)**

Returns the USBTMC information used by the USB hosts to determine the instrument interfaces.

**Group**

Miscellaneous

**Syntax**

USBTMC?

**USBTMC:PRODUCTID:DECimal? (Query Only)**

This query to returns the USBTMC product ID. The product ID numbers vary for each instrument family and model.

**Group**

Miscellaneous

**Syntax**

USBTMC:PRODUCTID:DECimal?
**USBTMC:PRODUCTID:HEXadecimal? (Query Only)**

This query returns the USBTMC product ID. The product ID numbers vary for each instrument family and model.

**Examples**

USBTMC:PRODUCTID:HEXadecimal? might return USBTMC:PRODUCTID:HEXadecimal 0x0401, indicating the product ID in hexadecimal is 0x0401.

**USBTMC:SERIALnumber? (Query Only)**

This query returns the instrument serial number.

**Examples**

USBTMC:SERIALNUMBER? might return USBTMC:SERIALNUMBER PQ30003, indicating the product serial number is PQ30003.
**USBTMC:VENDORID:DECimal? (Query Only)**

This query returns the USBTMC vendor ID. The Vendor ID for Tektronix is 0x699 or 1689 decimal.

**Group**
Miscellaneous

**Syntax**
USBTMC:VENDORID:DECimal?

**Related Commands**

**Examples**
USBTMC:VENDORID:DECIMAL? might return USBTMC:VENDORID:DECIMAL 1689, indicating the vendor ID in decimal is 1689.

**USBTMC:VENDORID:HEXadecimal? (Query Only)**

This query returns the USBTMC vendor ID. The Vendor ID for Tektronix is 0x699 or 1689 decimal.

**Group**
Miscellaneous

**Syntax**
USBTMC:VENDORID:HEXadecimal?

**Related Commands**

**Examples**
USBTMC:VENDORID:HEXADECIMAL? might return USBTMC:VENDORID:HEXADECIMAL 0x0699, indicating the vendor ID in hexadecimal is 0x0699.

**VERBose**

Sets or returns 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). This command does affect the verbose state of both the USBTMC and VXI-11 interfaces. Refer to the Introduction for information.

Group Miscellaneous

Syntax VERBose {OFF|ON|<NR1>}

Related Commands HEADer, *LRN?, SET?

Arguments 
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.

<NR1> a 0 returns minimum-length keywords for applicable setting queries; any other value returns full-length keywords.

Examples VERBOSE ON enables the Verbose state.

VERBOSE ? might return :VERB 0 indicating that the Verbose state is disabled.

*WAI (No Query Form)

Prevents the oscilloscope 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 oscilloscope with your application program (See page 3-7, Synchronization Methods.)

The *WAI command will stop execution until certain oscilloscope operations are complete. (See Table 3-3 on page 3-7.)

Group Status and Error

Syntax *WAI

Related Commands BUSY?, *OPC
Examples

*WAI prevents the oscilloscope from executing any further commands or queries until all pending commands that generate an OPC message are complete.

**WAVFrm? (Query Only)**

Returns WFMOutpre? and CURVe? data for the waveform as specified by the DATA:SOUrce command. This command is equivalent to sending both WFMOutpre? and CURVe?, with the additional provision that the response to WAVFrm? is guaranteed to provide a synchronized preamble and curve. The source waveform, as specified by :DATA:SOURCE, must be active or the query will not return any data and will generate an error indicator.

Group
Waveform Transfer

Syntax
WAVFrm?

Related Commands
CURVe, DATa:SOUrce, WFMOutpre?

Examples
WAVFRM? might return the waveform data as:

```
:WFMINPRE:BIT_NR 8;BN_FMT RI;BYT_NR 1; BYT_OR MSB;ENCDG ASCII;BN_FMT RI;BYT_NR 1;BIT_NR 8;ENCDG ASCII;BN_FMT RI;BYT_OR MSB;WFIGD "Ch1, DC coupling, 100.0mV/div, 4.000us/div, 10000 points, Sample mode";NR_PT 20;PT_FMT Y;XUNIT "s";XINCR 4.0000E-9;XZERO -20.0000E-6;PT_OFF 0;YUNIT "V";YMULT 4.0000E-3;YOFF 0.0000;YZERO 0.0000;:CURVE 2,1,4,2,4,3,0,3,3,3,3,3,3,4,3,5,6,6,7,3
```

**WFMINpre? (Query Only)**

Returns the waveform formatting and scaling specifications to be applied to the next incoming CURVe command data.

Group
Waveform Transfer

Syntax
WFMINpre?

Related Commands
WFMOutpre?

Examples
WFMINPRE? might return the waveform formatting as:

```
:WFMINPRE:BIT_NR 8;BN_FMT RI;BYT_NR 1; BYT_OR MSB;ENCDG BIN;NR_PT 500;PT_FMT
```

MSO4000 and DPO4000 Series Programmer Manual 2-509
\textbf{WFMInpre:BIT\_Nr}  

Sets or returns the number of bits per binary waveform point for the incoming waveform. Changing the value of \texttt{WFMInpre:BIT\_Nr} also changes the value of \texttt{WFMInpre:BYT\_Nr}.

\textbf{Group} \hspace{1em} Waveform Transfer

\textbf{Syntax} \hspace{1em} \texttt{WFMInpre:BIT\_Nr} <NR1>  
\hspace{1em} \texttt{WFMInpre:BIT\_Nr}\?  

\textbf{Related Commands} \hspace{1em} \texttt{WFMInpre:BYT\_Nr}

\textbf{Arguments} \hspace{1em} <NR1> number of bits per data point can be 8 or 16.

\textbf{Examples} \hspace{1em} \texttt{WFMINPRE:BIT\_NR 16} sets the number of bits per waveform point to 16, for incoming data.  
\hspace{1em} \texttt{WFMINPRE:BIT\_NR}\? might return \texttt{:WFMINPRE:BIT\_NR 8} indicating that incoming waveform data uses 8 bits per waveform point.

\textbf{WFMInpre:BN\_Fmt}  

Sets or returns the format of binary data for incoming waveforms.

\textbf{Group} \hspace{1em} Waveform Transfer

\textbf{Syntax} \hspace{1em} \texttt{WFMInpre:BN\_Fmt \{RI|RP\}}  
\hspace{1em} \texttt{WFMInpre:BN\_Fmt}\?  

\textbf{Related Commands} \hspace{1em} \texttt{WFMOutpre:BN\_Fmt}

\textbf{Arguments} \hspace{1em} RI specifies signed integer data point representation.  
\hspace{1em} RP specifies positive integer data point representation.
**Examples**

WFMINPRE:BN_FMT RP specifies positive integer data point representation.

WFMINPRE:BN_FMT? might return :WFMINPRE:BN_FMT RI indicating that the incoming data is currently interpreted as signed integers.

**WFMINpre:BYT_Nr**

Sets or returns the data width for the incoming waveform. Changing the value of WFMINpre:BYT_Nr also changes the value of WFMINpre:BIT_Nr.

**Group**

Waveform Transfer

**Syntax**

WFMINpre:BYT_Nr <NR1>
WFMINpre:BYT_Nr?

**Related Commands**

WFMINpre:BIT_Nr

**Arguments**

<NR1> is the number of bytes per data point and can be 1 or 2.

**Examples**

WFMINPRE:BYT_NR 1 sets the number of bytes per incoming waveform data point to 1, which is the default setting.

WFMINPRE:BYT_NR? might return :WFMINPRE:BYT_NR 2 indicating that there are 2 bytes per incoming waveform data point.

**WFMINpre:BYT_Or**

Sets or returns which byte of binary waveform data is expected first for incoming waveform data when data points require than one byte. This specification only has meaning when WFMINpre:ENCdg is set to BIN and WFMINpre:BYT_Nr is 2.

**Group**

Waveform Transfer

**Syntax**

WFMINpre:BYT_Or {LSB|MSB}
WFMINpre:BYT_Or?

**Related Commands**

WFMINpre:ENCdg, WFMINpre:BYT_Nr, WFMOutpre:BYT_Or
**Arguments**

- LSB specifies that the least significant byte will be expected first.
- MSB specifies that the most significant byte will be expected first.

**Examples**

- `WFMInpre:BYT_OR MSB` sets the most significant incoming byte of incoming waveform data to be expected first.
- `WFMInpre:BYT_OR?` might return `:WFMInpre:BYT_OR LSB` indicating that the least significant incoming CURVe data byte will be expected first.

---

**WFMInpre:ENCdg**

Sets or returns the type of encoding for incoming waveform data.

**Group**

Waveform Transfer

**Syntax**

- `WFMInpre:ENCdg {ASCII|Binary}`
- `WFMInpre:ENCdg?`

**Related Commands**

- `WFMOutpre:ENCdg`

**Arguments**

- ASCII specifies that the incoming data is in ASCII format.
- Binary specifies that the incoming data is in a binary format whose further interpretation requires knowledge of BYT_NR, BIT_NR, BN_FMT, and BYT_OR.

**Examples**

- `WFMInpre:ENCdg ASCII` sets the format of incoming waveform data to ASCII format.
- `WFMInpre:ENCdg?` might return `:WFMInpre:ENCdg Binary` indicating that the incoming waveform data is in binary format.

---

**WFMInpre:NR_Pt**

Sets or returns the number of data points that are in the incoming waveform record.

**Group**

Waveform Transfer

**Syntax**

- `WFMInpre:NR_Pt <NR1>`
- `WFMInpre:NR_Pt?`
Related Commands

Related Commands
CURVe, DATa, DATa:START, DATa:STOP, SAVe:WAVEform,
SAVe:WAVEform:FILEFormat, WFMOutpre:NR_Pt?

Arguments

Arguments
<NR1> is the number of data points if WFMInpre:PT_Fmt is set to Y. It is the
number of min-max pairs if WFMInpre:PT_Fmt is set to ENV.

Examples

Examples
WFMInpre:NR_PT 10000 specifies that 10000 data points will be expected.
WFMInpre:NR_PT ? might return :WFMInpre:NR_PT 10000 indicating that
there are 10000 data points in the expected incoming waveform record.

WFMInpre:PTFmt

WFMInpre:PTFmt
Sets or returns the point format of the incoming waveform data. Regardless of
the argument used, the scale, offset, and so on are interpreted similarly. When
ENV is used, waveform data is interpreted over the min-max pair; when Y is
used, it is interpreted over a single point.

Group
Waveform Transfer

Syntax

Syntax
WFMInpre:PTFmt {ENV|Y}
WFMInpre:PTFmt?

Related Commands

Related Commands
WFMOutpre:PTFmt?

Arguments

Arguments
ENV specifies that the waveform is transmitted in envelope mode as maximum
and minimum point pairs. Only Y values are explicitly transmitted. Absolute
coordinates are given by:

\[ X_n = X_{Zero} + X_{Inc}(n - PT_{Off}) \]
\[ Y_{nmax} = Y_{Zero} + Y_{Mult}(yn_{max} - Y_{Off}) \]
\[ Y_{nmin} = Y_{Zero} + Y_{Mult}(yn_{min} - Y_{Off}) \]

Y specifies a normal waveform where one ASCII or binary data point is
transmitted for each point in the waveform record. Only Y values are explicitly
transmitted. Absolute coordinates are given by:

\[ X_n = X_{Zero} + X_{Inc}(n - PT_{Off}) \]
\[ Y_n = Y_{Zero} + Y_{Mult}(yn - Y_{Off}) \]
**Examples**

WFMINPRE:PT_FMT ENV sets the incoming waveform data point format to enveloped.

WFMINPRE:PT_FMT? might return :WFMINPRE:PT_FMT ENV indicating that the waveform is transmitted as maximum and minimum point pairs.

**WFMINpre:PT_Off**

The set form of this command is ignored. The query form always returns a 0. This command is listed for compatibility with other Tektronix oscilloscopes.

**Group**

Waveform Transfer

**Syntax**

WFMINpre:PT_Off <NR1>

WFMINpre:PT_Off?

**Arguments**

Arguments are ignored.

**WFMINpre:XINcr**

Sets or returns the horizontal interval between incoming waveform points in units specified by WFMINpre:XUNit.

**Group**

Waveform Transfer

**Syntax**

WFMINpre:XINcr <NR3>

WFMINpre:XINcr?

**Related Commands**

WFMINpre:XUNit, WFMOutpre:XINcr?

**Arguments**

<NR3> is the horizontal interval representation.

**Examples**

WFMINPRE:XINCR 3E-3 sets the interval between incoming waveform points to 3 ms.

WFMINPRE:XINCR? might return :WFMINPRE:XINCR 1.0000E-3 indicating that if WFMINpre:XUNit is set to "s", there is a 1 ms interval between incoming waveform points.
WFMInpre:XUNit

Sets or returns the horizontal units of the incoming waveform.

Supported units are:
\%
Hz
A
A/A
A/W
A/\text{dB}
A/s
AA
AW
AdB
As
B
Hz
IRE
S/s
V
V/A
V/V
V/W
V/\text{dB}
V/s
VV
VW
VdB
Volts
Vs
W/A
W/V
W/W
W/\text{dB}
W/s
WA
WW
WdB
Ws
dB
dB/A
dB/V
dB/W
dB/db
dBAdB
dB/V
dB/dB
dBA
dBV
dBW
dBdB
day
degrees
div
hr
min
ohms
percent
s

Group
Waveform Transfer

Syntax
WFMInpre:XUNit <QString>
WFMInpre:XUNit?

Related Commands
WFMOutpre:XUNit?

Arguments
<QString> contains a maximum of three alpha characters that represent the horizontal unit of measure for the incoming waveform.

Examples
WFMInPRE:XUNIT "HZ" specifies that the horizontal units for the incoming waveform are hertz.

WFMInPRE:XUNIT? might return :WFMInPRE:XUNIT "s" indicating that the horizontal units for the incoming waveform are seconds.

WFMInpre:XZEro

Sets or returns the position value, in XUNits, of the first sample of the incoming waveform.

Group
Waveform Transfer

Syntax
WFMInpre:XZEro <NR3>
WFMInpre:XZEro?

Related Commands
WFMInpre:XINcr, WFMInpre:XUNit, WFMOutpre:XZEro?

Arguments
<NR3> argument is the floating point value of the position, in XUNits, of the first sample in the incoming waveform. If XUNits is “s”, <NR3> is the time of the first sample in the incoming waveform.
Examples

`WFMINPRE:XZERO 5.7E-6`, which indicates the trigger occurred 5.7 µs before the first sample in the waveform.

`WFMINPRE:XZERO?` might return `:WFMINPRE:XZERO -7.5000E-6` indicating that the trigger occurs 7.5 µs after the first sample in the waveform.

**WFMINpre:YMUlt**

Sets or returns the vertical scale factor of the incoming waveform, expressed in YUNits per waveform data point level. For one byte waveform data, there are 256 data point levels. For two byte waveform data there are 65,536 data point levels.

YMUlt, YOFF, and YZERO are used to convert waveform record values to YUNIT values using the following formula (where dl is the data level; curve_in_dl is a data point in CURVe):

\[
\text{value\_in\_units} = ((\text{curve\_in\_dl} - \text{YOFF\_in\_dl}) \times \text{YMUlt}) + \text{YZERO\_in\_units}
\]

**NOTE.** For a given waveform record, YMUlt, YOFF, and YZERO have to be a consistent set, otherwise vertical cursor readouts and vertical measurements may give incorrect results.

Group

Waveform Transfer

Syntax

`WFMINpre:YMUlt <NR3>`

`WFMINpre:YMUlt?`

Related Commands

DATa:DESTination, WFMINpre:BYT_Nr, WFMINpre:YUNIT

Arguments

`<NR3>` is the vertical scale factor per digitizing level of the incoming waveform points.

Examples

`WFMINpre:YMULT?` might return `:WFMINpre:YMULT 40.0000E-3`, which (if YUNIT is "V") indicates that the vertical scale is 40 mV/digitizing level (1V/div for 8-bit data).

`WFMINpre:YMULT 20E-3` specifies that (if WFMINpre:YUNIT is "V" and WFMINpre:BYT_Nr is 1), the vertical scale is 20 mV/digitizing level (500 mV/div).
**WFMInpre:YOFf**

Sets or returns the vertical position of the incoming waveform in digitizing levels. Variations in this number are analogous to changing the vertical position of the waveform.

YMUlt, YOFf, and YZEro are used to convert waveform record values to YUNit values using the following formula (where dl is the data level; curve_in_dl is a data point in CURve):

\[
\text{value\_in\_units} = ((\text{curve\_in\_dl} - \text{YOFf\_in\_dl}) \times \text{YMUlt}) + \text{YZero\_in\_units}
\]

**NOTE.** For a given waveform record, YMUlt, YOFf, and YZEO have to be a consistent set, otherwise vertical cursor readouts and vertical measurements may give incorrect results.

**Group**  
Waveform Transfer

**Syntax**  
WFMInpre:YOFf <NR3>  
WFMInpre:YOFf?

**Related Commands**  
WFMInpre:BYT_Nr, WFMInpre:YMUlt, WFMOutpre:YOFf?

**Arguments**  
<NR3> is the vertical offset in digitizing levels.

**Examples**  
WFMInpre:YOFf 50 specifies that the zero reference point for the incoming waveform is 50 digitizing levels (2 divisions, for 8-bit data) above the center of the data range.


**WFMInpre:YUNit**

Sets or returns the vertical units of the incoming waveform.

**Syntax**

WFMInpre:YUNIT <QString>

WFMInpre:YUNIT?

**Related Commands**

WFMOutpre:YUNIT?

**Arguments**

<QString> contains a maximum of three alpha characters that represent the vertical unit of measure for the incoming waveform.

**Examples**

WFMInpre:YUNIT? might return :WFMInpre:YUNIT "V" indicating the vertical units for the incoming waveform are volts.

WFMInpre:YUNIT "A" specifies that the vertical units for the incoming waveform are Amperes.

---

**WFMInpre:YZero**

Sets or returns the vertical offset of the incoming waveform in units specified by WFMInpre:YUNIT. Variations in this number are analogous to changing the vertical offset of the waveform.

YMUlt, YOFF, and YZERo are used to convert waveform record values to YUNIT values using the following formula (where dl is the data level; curve_in_dl is a data point in CURVe):

\[
\text{value\_in\_units} = ((\text{curve\_in\_dl} - \text{YOFF\_in\_dl}) \times \text{YMUlt}) + \text{YZERo\_in\_units}
\]

**NOTE.** For a given waveform record, YMUlt, YOFF, and YZERo have to be a consistent set, otherwise vertical cursor readouts and vertical measurements may give incorrect results.

**Group**

Waveform Transfer

**Syntax**

WFMInpre:YZERO <NR3>

WFMInpre:YZERO?

**Related Commands**

WFMInpre:YUNIT, WFMOutpre:YZERO?

**Arguments**

<NR3> is the offset in YUNITs.
**Examples**

`WFMINPRE:YZERO 1.5E+0` specifies that the zero reference point for the incoming waveform is 1.5 V below the center of the data range (given that `WFMINpre:YUNit` is set to V).

`WFMINPRE:YZERO?` might return `:WFMINPRE:YZEro 7.5000E-6` indicating that the zero reference for the incoming waveform is 7.5 µV below the center of the data range (given that `WFMINpre:YUNit` is set to V).

**WFMOOutpre? (Query Only)**

Returns waveform transmission and formatting parameters for the waveform specified by `DATa:SOUrce` command. If the waveform specified by the `DATa:SOUrce` command is not displayed, the oscilloscope returns only the waveform transmission parameters (`BYT_Nr`, `BIT_Nr`, `ENCdg`, `BN_Fmt`, `BYT_Or`).

**Group** Waveform Transfer

**Syntax** `WFMOOutpre?`

**Examples** `WFMOOUTPRE?` might return the waveform formatting data as:

`:WFMOOUTPRE:BYT_NR 2;BIT_NR 16;ENCDG ASCII;BN_FMT RI;BYT_OR MSB;WFID "Ch1, DC coupling, 100.0mV/div, 4.000us/div, 10000 points, Sample mode";NR_PT 10000;PT_FMT Y;XUNIT "s";XINCR 4.0000E-9;XZERO - 20.0000E-6;PT_OFF 0;YUNIT "V";YMULT 15.6250E-6;YOFF :"6.4000E+3;YZERO 0.0000`

**WFMOOutpre:BIT_Nr**

Sets and returns the number of bits per waveform point that outgoing waveforms contain, as specified by the `DATa:SOUrce` command. Changing the value of `WFMOOutpre:BIT_Nr` also changes the values of `WFMOOutpre:BYT_Or` and .

**Group** Waveform Transfer

**Syntax** `WFMOOutpre:BIT_Nr <NR1>`

`WFMOOutpre:BIT_Nr?`

**Related Commands** `DATa:SOUrce`, `WFMOOutpre:BN_Fmt`

**Arguments** `<NR1>` is the number of bits per data point and can be 8 or 16.
Examples

WFMOUTPRE:BIT_NR 16 sets the number of bits per waveform point to 16 for outgoing waveforms.

WFMOUTPRE:BIT_NR? might return :WFMOUTPRE:BIT_NR 8 indicating that outgoing waveforms use 8 bits per waveform point.

WFMOutpre:BN_Fmt

Sets or returns the format of binary data for outgoing waveforms specified by the DATa:SOUrce command. Changing the value of WFMOutpre:BN_Fmt also changes the value of DATa:ENCdg.

Group Waveform Transfer

Syntax WFMOutpre:BN_Fmt {RI|RP}
WFMOutpre:BN_Fmt?

Related Commands DATa:ENCdg, DATa:SOUrce

Arguments RI specifies signed integer data point representation.
RP specifies positive integer data point representation.

Examples WFMOUTPRE:BN_FMT RP specifies that outgoing waveform data will be in positive integer 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

Sets or returns the data width for the outgoing waveform specified by the DATa:SOUrce command. Changing WFMOutpre:BYT_Nr also changes WFMOutpre:BIT_Nr and .

Group Waveform Transfer

Syntax WFMOutpre:BYT_Nr <NR1>
WFMOutpre:BYT_Nr?

Related Commands DATa:SOUrce, , WFMOutpre:BIT_Nr
Arguments  

<NR1> is the number of bytes per data point and can be 1 or 2.

Examples  

WFMOUTPRE:BYT_NR 1 sets the number of bytes per outgoing waveform data point to 1, which is the default setting.
WFMOUTPRE:BYT_NR? might return :WFMOUTPRE:BYT_NR 2 indicating that there are 2 bytes per outgoing waveform data point.

WFMOutpre:BYT_Or  

Sets or returns which byte of binary waveform data is transmitted first, during a waveform data transfer, when data points require than one byte. This specification only has meaning when WFMOutpre:ENCdg is set to BIN and WFMOutpre:BYT_Nr is 2. Changing WFMOutpre:BYT_Or also changes DATa:ENCdg (if DATa:ENCdg is not ASCII).

Group  

Waveform Transfer

Syntax  

WFMOutpre:BYT_Or {LSB|MSB}  
WFMOutpre:BYT_Or?

Related Commands  

WFMOutpre:ENCdg, WFMOutpre:BYT_Nr

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:ENCdg  

Sets and queries the type of encoding for outgoing waveforms.

Group  

Waveform Transfer

Syntax  

WFMOutpre:ENCdg {ASCII|BINARY}  
WFMOutpre:ENCdg?
Commands Listed in Alphabetical Order

**Related Commands**
DATa:ENCdg, WFMOutpre:BYT_Nr, WFMOutpre:BYT_Or, WFMOutpre:BIT_Nr, WFMOutpre:BN_Fmt

**Arguments**

**ASCi** specifies that the outgoing data is to be in ASCII format. Waveforms will be sent as \(<NR1>\) 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? might return :WFMOUTPRE:ENCDG BIN indicating that outgoing waveform data will be sent in binary format.

WFMOUTPRE:ENCDG ASC specifies that the outgoing waveform data will be sent in ASCII format.

---

**WFMOutpre:FRACTional? (Query Only)**

The set form of this command is ignored. The query form always returns a 0, if the waveform specified by DATA:SOUrce is on or displayed. If the waveform is not displayed, the query form generates an error and returns event code 2244. This command is for compatibility with other Tektronix oscilloscopes.

**Group**
Waveform Transfer

**Syntax**
WFMOutpre:FRACTional?

**Related Commands**
DATa:SOUrce

**Arguments**
Arguments are ignored.

---

**WFMOutpre:NR_Pt? (Query Only)**

Returns the number of points for the DATa:SOUrce waveform that will be transmitted in response to a CURVe? query. The query command will time out and an error will be generated if the waveform specified by DATa:SOUrce is not turned on.

**Group**
Waveform Transfer
Syntax

WFMOutpre:NR_Pt?

Related Commands

CURVe, DATa, DATa:STARt, DATa:STOP, SAVe:WAVEform, SAVe:WAVEform:FILEFormat, WFMInpre:NR_Pt

Examples

WFMOutpre:NR_Pt? might return :WFMOutpre:NR_Pt 10000 indicating that there are 10000 data points to be sent.

WFMOutpre:PT_Fmt? (Query Only)

Returns the point format for the outgoing waveform specified by the DATa:SOUrce command. Returned values are either ENV, which indicates envelope mode format in which the data is returned as a series of min/max pairs, or Y, which indicates normal waveform points. The query command will time out and an error will be generated if the waveform specified by DATa:SOUrce is not turned on.

Group

Waveform Transfer

Syntax

WFMOutpre:PT_Fmt?

Related Commands

CURVe, DATa:SOUrce

Examples

WFMOutpre:PT_FMT? might return :WFMOutpre:PT_Fmt ENV indicating that the waveform data is a series of min-max pairs.

WFMOutpre:PT_Off? (Query Only)

The set form of this command is ignored. The query form always returns a 0, if the waveform specified by DATA:SOUrce is on or displayed. If the waveform is not displayed, the query form generates an error and returns event code 2244. This command is for compatibility with other Tektronix oscilloscopes.

Group

Waveform Transfer

Syntax

WFMOutpre:PT_Off?

Related Commands

DATa:SOUrce
Arguments  
Arguments are ignored.

Examples  
WFMOUTPRE:PT_OFF? might return WFMOUTPRE:PT_OFF 0 indicating that the waveform specified by DATA:SOURce is on or displayed.

WFMOutpre:PT_ORDER? (Query Only)

This query is for compatibility with other Tektronix oscilloscopes and always returns LINEAR.

Group  
Waveform Transfer

Syntax  
WFMOutpre:PT_ORDER?

Related Commands  
DATa:SOURce

Examples  
WFMOUTPRE:PT_ORDER? returns :WFMOUTPRE:PT_ORDER LINEAR.

WFMOutpre:WFId? (Query Only)

Returns a string describing several aspects of the acquisition parameters for the waveform specified by the DATa:SOURce command. The query command will time out and an error will be generated if the waveform specified by DATa:SOURce is not turned on.

Group  
Waveform Transfer

Syntax  
WFMOutpre:WFId?

Related Commands  
DATa:SOURce

Returns  
QString comprises the following comma-separated fields documented in the tables below:
### Table 2-45: 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;CH1–4&quot; &quot;Math1 &quot;Ref1–4&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>
<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;50.0.0 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;</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;1000 points&quot; &quot;1000000 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;Envelope mode&quot; &quot;Average mode&quot;</td>
</tr>
</tbody>
</table>

**Examples**

WFMOUTPRE:WFID? might return:WFMOUTPRE:WFID "Ch1, DC coupling, 100.0 mV/div, 500.0 μs/div, 1000 points, Sample mode"

**WFMOutpre:XINcr? (Query Only)**

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. The query command will time out and an error will be generated if the waveform specified by DATa:SOUrce is not turned on.

**Group** Waveform Transfer

**Syntax** WFMOutpre:XINcr?
Related Commands  

**Related Commands**  
DATa:SOUrce, WFMOutpre:XUNit?

Examples  

**Examples**  
WFMOUTPRE:XINCR? might return :WFMOUTPRE:XINCR 10.0000E-6 indicating that the horizontal sampling interval is 10 µs/point.

### WFMOutpre:XUNit? (Query Only)

Returns the horizontal units for the waveform specified by the DATa:SOUrce command. The query command will time out and an error will be generated if the waveform specified by DATa:SOUrce is not turned on.

**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:XZEro? (Query Only)

Returns the time coordinate of the first point in the outgoing waveform.

This value is in units of WFMOutpre:XUNit?. The query command will time out and an error will be generated if the waveform specified by DATa:SOUrce is not turned on.

**Group**  
Waveform Transfer

**Syntax**  
WFMOutpre:XZEro?

**Related Commands**  
DATa:SOUrce, WFMOutpre:XUNit?

**Examples**  
WFMOUTPRE:XZERO? might return :WFMOUTPRE:XZERO 5.6300E-9 indicating that the trigger occurred 5.63 ns before the first sample in the waveform record.
WFMOutpre:YMUlt? (Query Only)

Returns the vertical scale factor per digitizing level in units specified by WFMOutpre:YUNit for the waveform specified by the DATa:SOUrce command. The query command will time out and an error is generated if the waveform specified by DATa:SOUrce is not turned on.

See the description of WFMInpre:YMUlt to see how this scale factor is used to convert waveform sample values to volts.

Group    Waveform Transfer
Syntax   WFMOutpre:YMUlt?
Related Commands    DATa:SOUrce, WFMInpre:YMUlt
Examples    WFMOUTPRE:YMULT? might return :WFMOUTPRE:YMULT 4.0000E-3 indicating that the vertical scale for the corresponding waveform is 100 mV/div (for 8-bit waveform data).

WFMOutpre:YOFF? (Query Only)

Returns the vertical position in digitizing levels for the waveform specified by the DATa:SOUrce command. The query command will time out and an error will be generated if the waveform specified by DATa:SOUrce is not turned on.

See the description of WFMInpre:YOFF to see how this position is used to convert waveform sample values to volts.

Group    Waveform Transfer
Syntax   WFMOutpre:YOFF?
Related Commands    DATa:SOUrce, WFMOutpre:BYT_Nr
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 (for 8-bit waveform data).
WFMOOutpre:YUNit? (Query Only)

Returns the vertical units for the waveform specified by the DATa:SOUrce command. The query command will time out and an error will be generated if the waveform specified by DATa:SOUrce is not turned on.

Group   Waveform Transfer

Syntax   WFMOOutpre:YUNit?

Related Commands   DATa:SOUrce

Examples   WFMOOUTPRE:YUNIT? might return :WFMOOUTPRE:YUNIT "dB" indicating that the vertical units for the waveform are measured in decibels.

WFMOOutpre:YZEro? (Query Only)

Returns the vertical offset in units specified by WFMOOutpre:YUNit? for the waveform specified by the DATa:SOUrce command. The query command will time out and an error will be generated if the waveform specified by DATa:SOUrce is not turned on.

See the description of WFMIInpre:YZEro to see how this offset is used to convert waveform sample values to volts.

Group   Waveform Transfer

Syntax   WFMOOutpre:YZEro?

Related Commands   DATa:SOUrce, WFMOOutpre:YUNit?

Examples   WFMOOUTPRE:YZERO? might return :WFMOOUTPRE:YZERO -100.0000E-3 indicating that vertical offset is set to -100 mV.

ZOOm? (Query Only)

Returns the current vertical and horizontal positioning and scaling of the display.

Group   Zoom
### Syntax

ZOom?

### Examples

ZOom? might return:

```
:ZOOM:MODE 1;GRATICULE:SIZE 80;SPLIT EIGHTYTWENTY;:ZOOM:ZOOM1:STATE 1;SCALE 400.0000E-12;POSITION 46.8986;FACTOR 50.0000E+3;HORIZONTAL:POSITION 46.8986;SCALE 40.0000E-12
```

### ZOom:MODe

Turns Zoom mode on or off. The Zoom query returns the current state of Zoom mode. This command is equivalent to pressing the zoom button located on the front panel.

#### Group

Zoom

#### Syntax

ZOom:MODe {ON|OFF}<NR1>

#### Arguments

- **ON** turns on Zoom mode.
- **OFF** turns off Zoom mode.
- **<NR1>** = 0 turns off Zoom mode; any other value turns on Zoom mode.

#### Examples

ZOom:MODE OFF turns off Zoom mode.

ZOom:MODE? might return :ZOOM:MODE 1 indicating that Zoom mode is currently turned on.

### ZOom:ZOOM<x>? (Query Only)

Returns the current vertical and horizontal positioning and scaling of the display. **<x>** can only be 1.

#### Group

Zoom

#### Syntax

ZOom:ZOOM<x>?
Examples

ZOOM:ZOOM1? might return :ZOOM:ZOOM1:STATE 1;SCALE 400.0000E-12;POSITION 46.8986;FACTOR 50.0000E+3;HORIZONTAL:POSITION 46.8986;SCALE 400.0000E-12.

**ZOOM:ZOOM<x>:FACTOR? (Query Only)**

Returns the zoom factor of a particular zoom box. <x> can only be 1.

**Group**

Zoom

**Syntax**

ZOOM:ZOOM<x>:FACTOR?

**Returns**

<NR1> is the zoom factor of a zoom box.

**ZOOM:ZOOM<x>:POSITION**

Sets the horizontal position of the zoom box, in terms of 0 to 100.0% of upper window. <x> can only be 1.

**Group**

Zoom

**Syntax**

ZOOM:ZOOM<x>:POSITION <NR3>

ZOOM:ZOOM<x>:POSITION?

**Arguments**

<NR3> is the horizontal position as a percent of the upper window.

**ZOOM:ZOOM<x>:SCALE**

Sets or returns the horizontal scale of the zoom box. <x> can only be 1.

**Group**

Zoom

**Syntax**

ZOOM:ZOOM<x>:SCALE <NR3>

ZOOM:ZOOM<x>:SCALE?

**Arguments**

<NR3> is the horizontal scale of the zoom box.
**ZOOM:ZOOM<x>:STATE**

Sets or returns the specified zoom on or off, where x is the integer 1 representing the single zoom window. <x> can only be 1.

**Group**  
Zoom

**Syntax**  
ZOOM:ZOOM<x>:STATE {ON|OFF|<NR1>}
ZOOM:ZOOM<x>:STATE?

**Arguments**  
ON turns Zoom 1 on.

OFF turns Zoom 1 off.

<NR1> = 0 disables the specified zoom; any other value enables the specified zoom.

**Examples**  
ZOOM:ZOOM1:STATE ON turns Zoom1 on.

ZOOM:ZOOM1:STATE? might return :ZOOM:ZOOM1:STATE 1 indicating that Zoom1 is on.
Status and Events

The oscilloscope provides a status and event reporting system for the Ethernet, GPIB (with the TEK-USB-488 Adapter), 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>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>
<tbody>
<tr>
<td>PON</td>
<td>URQ</td>
<td>CME</td>
<td>EXE</td>
<td>DDE</td>
<td>QYE</td>
<td>RQC</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><strong>PON</strong> Power On. Shows that the oscilloscope was powered on. On completion, the diagnostic self tests also set this bit.</td>
</tr>
<tr>
<td>6</td>
<td><strong>URQ</strong> User Request. Indicates that an application event has occurred. <em>See note.</em></td>
</tr>
<tr>
<td>5</td>
<td><strong>CME</strong> Command Error. Shows that an error occurred while the oscilloscope was parsing a command or query.</td>
</tr>
<tr>
<td>4</td>
<td><strong>EXE</strong> Execution Error. Shows that an error executing a command or query.</td>
</tr>
<tr>
<td>3</td>
<td><strong>DDE</strong> Device Error. Shows that a device error occurred.</td>
</tr>
<tr>
<td>2</td>
<td><strong>QYE</strong> Query Error. Either an attempt was made to read the Output Queue when no data was present or pending, or that data in the Output Queue was lost.</td>
</tr>
<tr>
<td>1</td>
<td><strong>RQC</strong> Request Control. This is not used.</td>
</tr>
<tr>
<td>0 (LSB)</td>
<td><strong>OPC</strong> Operation Complete. Shows that the operation is complete. This bit is set when all pending operations complete following an *OPC command.</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><strong>RQS</strong> Request Service. Obtained from a serial poll. Shows that the oscilloscope requests service from the GPIB controller.</td>
</tr>
<tr>
<td>6</td>
<td><strong>MSS</strong> Master Status Summary. Obtained from *STB? query. Summarizes the ESB and MAV bits in the SBR.</td>
</tr>
<tr>
<td>5</td>
<td><strong>ESB</strong> Event Status Bit. Shows that status is enabled and present in the SESR.</td>
</tr>
</tbody>
</table>

---

3-2  
MSO4000 and DPO4000 Series Programmer Manual
**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.

---

**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</td>
</tr>
<tr>
<td>3</td>
<td>————</td>
</tr>
<tr>
<td>2</td>
<td>————</td>
</tr>
<tr>
<td>1–0</td>
<td>————</td>
</tr>
</tbody>
</table>
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.

![Figure 3-5: The Service Request Enable Register (SRER)](image)

### *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.
The Event Queue stores detailed information on up to 33 events. If more 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.

The figure below 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 process, the oscilloscope will continue processing other commands. This means that some operations will not be completed in the order that they were sent.

Sometimes the result of an operation depends on the result of an earlier operation. A first operation must complete before the next one is processed. The oscilloscope status and event reporting system is designed to accommodate this process.

The Operation Complete (OPC) bit of the Standard Event Status Register (SESR) can be programmed to indicate when certain oscilloscope operations have completed and, by setting the Event Status Enable Register (ESER) to report OPC in the Event Status Bit (ESB) of the Status Byte Register (SBR) and setting the Service Request Enable Register (SRER) to generate service request upon a positive transition of the ESB, a service request (SRQ) interrupt can be generated when certain operations complete as described in this section.

The following oscilloscope operations can generate an 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>Only when in single sequence acquisition mode</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:DEGAUss EXECute</td>
<td></td>
</tr>
<tr>
<td>AUXin: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>HARDCopy</td>
<td></td>
</tr>
<tr>
<td>HARDCopy START</td>
<td></td>
</tr>
<tr>
<td>RECALL:SETUp &lt;file as quoted string&gt;</td>
<td></td>
</tr>
<tr>
<td>RECALL:WAVEform &lt;file as quoted string&gt;</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;file as quoted string&gt;</td>
<td></td>
</tr>
</tbody>
</table>
Table 3-3: Oscilloscope operations that can generate OPC (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVe:WAVEform REF1</td>
<td>REF2</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
SELECT:CH1 ON
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
/** Take amplitude measurement **/
MEASUREMENT:MEAS1:VALUE?
```

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](image)

**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.

![Figure 3-8: Processing Sequence With Synchronization](image)

**Figure 3-8: Processing Sequence With Synchronization**
You can use four commands to synchronize the operation of the oscilloscope with your application program: *WAI, BUSY, *OPC, and *OPC?

**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
SELECT:CH1 ON
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*/
/*
*WAI
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE?
```

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.

**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
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Acquire waveform data */
```
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.

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:

/* Set up conditional acquisition */
ACQUIRE:STATE OFF
SELECT:CH1 ON
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 */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* Wait until the acquisition is complete before taking the measurement */
*OPC
while serial poll = 0, keep looping
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE?

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

/* Set up conditional acquisition */
ACQUIRE:STATE OFF
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Enable the status registers */
DESE 1
*ESE 1
*SRE 32
/* 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*/
*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:IMMED:VALUE?

**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
SELECT:CH1 ON
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*/
*OPC?

Wait for read from Output Queue.
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE?

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.

**Messages**
The information contained in the topic tabs 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 3-5: Command Error Messages (CME Bit 5)

<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>
<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>
</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>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>270</td>
<td>Hardcopy error</td>
</tr>
<tr>
<td>271</td>
<td>Hardcopy device not responding</td>
</tr>
<tr>
<td>272</td>
<td>Hardcopy is busy</td>
</tr>
<tr>
<td>273</td>
<td>Hardcopy aborted</td>
</tr>
<tr>
<td>274</td>
<td>Hardcopy configuration error</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>Print server not found</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>
<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>Code</td>
<td>Message</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------------</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>
<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>
</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>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 3-7: Device Error Messages (DDE Bit 3)

<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 3-8: System Event Messages

<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>
<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 a 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>
<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>CONTROL</td>
<td>NUMBERS</td>
<td>SYMBOLS</td>
<td>UPPER CASE</td>
<td>LOWER CASE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>NUL</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>0</td>
<td>DLE</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>0</td>
<td>SOH</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>0</td>
<td>ETX</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>0</td>
<td>EOT</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>0</td>
<td>SPC</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>0</td>
<td>NUM</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>0</td>
<td>LT</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>0</td>
<td>RTR</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>0</td>
<td>BLS</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>0</td>
<td>SHL</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>0</td>
<td>LF</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>0</td>
<td>VT</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>0</td>
<td>CR</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>0</td>
<td>SO</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>0</td>
<td>SI</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>0</td>
<td>ADDRESS</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>0</td>
<td>ADDRESS</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>0</td>
<td>ADDRESS</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>0</td>
<td>ADDRESS</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
</tr>
<tr>
<td>0</td>
<td>ADDRESS</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>0</td>
<td>ADDRESS</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
</tr>
</tbody>
</table>

**KEY**
- **octal**: 5
- **hex**: 5
- **ENQ**: 5
- **PPC**: ASCII character with ATN asserted
- **decimal**: 5

**GPIB command**: Ref: ANSI STD X3.4-1977
**IEEE STD**: 488.1-1987
**ISO STD**: 646-2973

---

**MSO4000 and DPO4000 Series Programmer Manual**

A-1
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>Spellings</th>
</tr>
</thead>
<tbody>
<tr>
<td>*CAL</td>
<td>ADVanced</td>
</tr>
<tr>
<td>*CLS</td>
<td>ALIas</td>
</tr>
<tr>
<td>*DDT</td>
<td>ALL</td>
</tr>
<tr>
<td>*ESE</td>
<td>ALLev</td>
</tr>
<tr>
<td>*ESR</td>
<td>ALLFields</td>
</tr>
<tr>
<td>*IDN</td>
<td>ALLLines</td>
</tr>
<tr>
<td>*LRN</td>
<td>ALLQString</td>
</tr>
<tr>
<td>*OPC</td>
<td>ALTERNATE</td>
</tr>
<tr>
<td>*PSC</td>
<td>ALWAYS</td>
</tr>
<tr>
<td>*PUD</td>
<td>AMPlitude</td>
</tr>
<tr>
<td>*RCL</td>
<td>AMPSVIAVOLts</td>
</tr>
<tr>
<td>*RST</td>
<td>AND</td>
</tr>
<tr>
<td>*SAV</td>
<td>ANY</td>
</tr>
<tr>
<td>*SRE</td>
<td>APPKey</td>
</tr>
<tr>
<td>*STB</td>
<td>APPpwr</td>
</tr>
<tr>
<td>*TRG</td>
<td>AREA</td>
</tr>
<tr>
<td>*TST</td>
<td>ASCII</td>
</tr>
<tr>
<td>*WAI</td>
<td>ASSIgn</td>
</tr>
<tr>
<td>1CHx</td>
<td>ATRIGger</td>
</tr>
<tr>
<td>1NR3</td>
<td>AUDIO</td>
</tr>
<tr>
<td>A</td>
<td>AUTO</td>
</tr>
<tr>
<td>A0</td>
<td>AUTOSet</td>
</tr>
<tr>
<td>A1</td>
<td>AUTOZero</td>
</tr>
<tr>
<td>A2</td>
<td>AUX</td>
</tr>
<tr>
<td>A3</td>
<td>AUXOut</td>
</tr>
<tr>
<td>A4</td>
<td>AUXin</td>
</tr>
<tr>
<td>A5</td>
<td>AVERAGE</td>
</tr>
<tr>
<td>A6</td>
<td>Auto</td>
</tr>
<tr>
<td>A7</td>
<td>B</td>
</tr>
<tr>
<td>A8</td>
<td>B0</td>
</tr>
<tr>
<td>A9</td>
<td>B1</td>
</tr>
<tr>
<td>ABORT</td>
<td>B2</td>
</tr>
<tr>
<td>ABORT</td>
<td>B3</td>
</tr>
<tr>
<td>ABSolute</td>
<td>B4</td>
</tr>
<tr>
<td>AC</td>
<td>B5</td>
</tr>
<tr>
<td>ACK</td>
<td>B6</td>
</tr>
<tr>
<td>ACKMISS</td>
<td>B7</td>
</tr>
<tr>
<td>ACQ</td>
<td>B8</td>
</tr>
<tr>
<td>ACQLENGTH</td>
<td>B9</td>
</tr>
<tr>
<td>ACQuire</td>
<td>BACKLight</td>
</tr>
<tr>
<td>ACTIVEprinter</td>
<td>BACKwards</td>
</tr>
<tr>
<td>ADD</td>
<td>BANDwidth</td>
</tr>
<tr>
<td>ADDR10</td>
<td>BASE</td>
</tr>
<tr>
<td>ADDR7</td>
<td>BDIFFBP</td>
</tr>
<tr>
<td>ADDRANDDATA</td>
<td>BINary</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>BITDelay</td>
</tr>
<tr>
<td>BITOrder</td>
<td>CM10BY15</td>
</tr>
<tr>
<td>BITRate</td>
<td>CM13BY18</td>
</tr>
<tr>
<td>BITSTUFFing</td>
<td>CM15BY21</td>
</tr>
<tr>
<td>BIT_Nr</td>
<td>CM18BY24</td>
</tr>
<tr>
<td>BIT</td>
<td>CM6BY8</td>
</tr>
<tr>
<td>BLAckmanharris</td>
<td>CM7BY10</td>
</tr>
<tr>
<td>BM</td>
<td>CM9BY13</td>
</tr>
<tr>
<td>BMP</td>
<td>CMEan</td>
</tr>
<tr>
<td>BN_Fmt</td>
<td>COLUMN</td>
</tr>
<tr>
<td>BOTH</td>
<td>COMMAND</td>
</tr>
<tr>
<td>BOX</td>
<td>CONDCALCmethod</td>
</tr>
<tr>
<td>BOXpct</td>
<td>CONDITION</td>
</tr>
<tr>
<td>BTRIGger</td>
<td>CONduction</td>
</tr>
<tr>
<td>BURst</td>
<td>CONFIGure</td>
</tr>
<tr>
<td>BUS</td>
<td>CONTROL</td>
</tr>
<tr>
<td>BUSY</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>BY</td>
<td>COPY</td>
</tr>
<tr>
<td>BYPpass</td>
<td>COUNT</td>
</tr>
<tr>
<td>BYT_Nr</td>
<td>COPy</td>
</tr>
<tr>
<td>BYT_Or</td>
<td>CPU</td>
</tr>
<tr>
<td>Block</td>
<td>CR</td>
</tr>
<tr>
<td>CAN</td>
<td>CRC</td>
</tr>
<tr>
<td>CANH</td>
<td>CRC16</td>
</tr>
<tr>
<td>CRMS</td>
<td>CRC5</td>
</tr>
<tr>
<td>CROSSHair</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>CREATE</td>
<td>CRCTRailer</td>
</tr>
<tr>
<td>CNAL</td>
<td>CREATE</td>
</tr>
<tr>
<td>CARD</td>
<td>CREATE</td>
</tr>
<tr>
<td>CURRENTSource</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>CREA</td>
<td>CRCTRailer</td>
</tr>
<tr>
<td>CURRent</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>CAtalog</td>
<td>CRCTRailer</td>
</tr>
<tr>
<td>CURSors</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>CH1</td>
<td>CRCTRailer</td>
</tr>
<tr>
<td>CURVes</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>CH3</td>
<td>CRCTRailer</td>
</tr>
<tr>
<td>CUSTom</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>CH4</td>
<td>CRCTRailer</td>
</tr>
<tr>
<td>CHannel</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>Checksum</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>CLASSALIMIT</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>CLASS</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>CLEAR</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>CLEARMenu</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>CLEARSNapshot</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>Clear</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>CLOCK</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>D0</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>D1</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>D10</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>D11</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>D12</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>D13</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>D14</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>D15</td>
<td>CRCHeader</td>
</tr>
<tr>
<td>Reserved Word</td>
<td>Reserved Word</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>D2</td>
<td>DYNAMIC</td>
</tr>
<tr>
<td>D3</td>
<td>Dx</td>
</tr>
<tr>
<td>D4</td>
<td>E</td>
</tr>
<tr>
<td>D5</td>
<td>ECL</td>
</tr>
<tr>
<td>D6</td>
<td>EDGE</td>
</tr>
<tr>
<td>D7</td>
<td>EEPROM</td>
</tr>
<tr>
<td>D8</td>
<td>EITHER</td>
</tr>
<tr>
<td>D9</td>
<td>ENABLE</td>
</tr>
<tr>
<td>DATA</td>
<td>ENCDg</td>
</tr>
<tr>
<td>DATA0</td>
<td>END</td>
</tr>
<tr>
<td>DATA1</td>
<td>ENDPoint</td>
</tr>
<tr>
<td>DATAPacket</td>
<td>ENERGY</td>
</tr>
<tr>
<td>DATE</td>
<td>ENET</td>
</tr>
<tr>
<td>DATEPrint</td>
<td>ENGLISH</td>
</tr>
<tr>
<td>DAll</td>
<td>ENV</td>
</tr>
<tr>
<td>DB</td>
<td>ENVELOPE</td>
</tr>
<tr>
<td>DC</td>
<td>EOF</td>
</tr>
<tr>
<td>DDT</td>
<td>EOFTYPE</td>
</tr>
<tr>
<td>DECimal</td>
<td>EOP</td>
</tr>
<tr>
<td>DEFLT</td>
<td>EQUAL</td>
</tr>
<tr>
<td>DEFAult</td>
<td>ERROR</td>
</tr>
<tr>
<td>DEFINE</td>
<td>ERRORTYPE</td>
</tr>
<tr>
<td>DEGAUss</td>
<td>ERRTYPE</td>
</tr>
<tr>
<td>DEGREES</td>
<td>ERRlog</td>
</tr>
<tr>
<td>DELEte</td>
<td>ETHERnet</td>
</tr>
<tr>
<td>DELIMITER</td>
<td>EVEN</td>
</tr>
<tr>
<td>DELTa</td>
<td>EVENT</td>
</tr>
<tr>
<td>DELTAtime</td>
<td>EVENTS</td>
</tr>
<tr>
<td>DELay</td>
<td>EVENTtable</td>
</tr>
<tr>
<td>DELayed</td>
<td>EVMsg</td>
</tr>
<tr>
<td>DESE</td>
<td>EVQty</td>
</tr>
<tr>
<td>DESKew</td>
<td>EXECute</td>
</tr>
<tr>
<td>DESTination</td>
<td>EXT</td>
</tr>
<tr>
<td>DHCPbootp</td>
<td>EXTended</td>
</tr>
<tr>
<td>DIAG</td>
<td>EXTERNAL</td>
</tr>
<tr>
<td>DIFFerential</td>
<td>FACTor</td>
</tr>
<tr>
<td>DIGital</td>
<td>FACTory</td>
</tr>
<tr>
<td>DIR</td>
<td>FAIL</td>
</tr>
<tr>
<td>DIREction</td>
<td>FAILures</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>FALL</td>
</tr>
<tr>
<td>DISABLEd</td>
<td>FALSE</td>
</tr>
<tr>
<td>Display</td>
<td>FALLing</td>
</tr>
<tr>
<td>Displaymode</td>
<td>FASTPHOTO</td>
</tr>
<tr>
<td>DMINus</td>
<td>FASTER</td>
</tr>
<tr>
<td>DNS</td>
<td>FASTest</td>
</tr>
<tr>
<td>DOMAINname</td>
<td>FFT</td>
</tr>
<tr>
<td>DOTsonly</td>
<td>FIELD</td>
</tr>
<tr>
<td>DPLUS</td>
<td>FIFTy</td>
</tr>
<tr>
<td>DPOModels</td>
<td>FILEFormat</td>
</tr>
<tr>
<td>DRAFT</td>
<td>FILESystem</td>
</tr>
<tr>
<td>DUAL</td>
<td>FILTER</td>
</tr>
<tr>
<td>DUMP</td>
<td>FINE</td>
</tr>
<tr>
<td>Reserved Words</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>Appendices</td>
<td>Reserved Words</td>
</tr>
<tr>
<td>Appendix B</td>
<td>Reserved Words</td>
</tr>
<tr>
<td>INDICATORS</td>
<td>LOW, NAK, NAME</td>
</tr>
<tr>
<td>INDIVidual</td>
<td>LOWCurrent, NAME</td>
</tr>
<tr>
<td>INDependent</td>
<td>LOWLimit, NAnd</td>
</tr>
<tr>
<td>INFINITE</td>
<td>LOWVoltage, NEgative</td>
</tr>
<tr>
<td>INIT</td>
<td>LOWerthreshold, NEWpass</td>
</tr>
<tr>
<td>INKSaver</td>
<td>LSB, NEXT</td>
</tr>
<tr>
<td>INPUTPOWER</td>
<td>MAG, NEXT</td>
</tr>
<tr>
<td>Input</td>
<td>MAGnivu, NEXT</td>
</tr>
<tr>
<td>INTENSITY</td>
<td>MAIN, NO</td>
</tr>
<tr>
<td>INTERLaced</td>
<td>MARK, NOCARE</td>
</tr>
<tr>
<td>INTERNAL</td>
<td>MARKSINColumn, NOISErej</td>
</tr>
<tr>
<td>INVERTed</td>
<td>MASK, NONE</td>
</tr>
<tr>
<td>INVert</td>
<td>MATH, NONE</td>
</tr>
<tr>
<td>INVerted</td>
<td>MATH1, NOPARity</td>
</tr>
<tr>
<td>INrange</td>
<td>MATHVAR, NOR</td>
</tr>
<tr>
<td>IO</td>
<td>MAX, NORMAL</td>
</tr>
<tr>
<td>IPADDRESS</td>
<td>MAXAmps, NOTCOMPuted</td>
</tr>
<tr>
<td>IRMS</td>
<td>MAXBytedelim, NR_HARMonics</td>
</tr>
<tr>
<td>ISCLOCKed</td>
<td>MAXSampleRate, NR_PT</td>
</tr>
<tr>
<td>ITALian</td>
<td>MAXVolts, NTIMES</td>
</tr>
<tr>
<td>JAPAnese</td>
<td>MAXWatts, PLOTTyPe</td>
</tr>
<tr>
<td>KOREan</td>
<td>MAXimum, NULL</td>
</tr>
<tr>
<td>K</td>
<td>MEAN, NULL</td>
</tr>
<tr>
<td>L2</td>
<td>MEAS, NULLFRDynamic</td>
</tr>
<tr>
<td>L4</td>
<td>MEASUREment, NULLFRStatic</td>
</tr>
<tr>
<td>LABEL</td>
<td>MEDIum, NULLFRStatic</td>
</tr>
<tr>
<td>LANGUAGE</td>
<td>MEG, NUMACq</td>
</tr>
<tr>
<td>LANDscape</td>
<td>MESSAGE, NUMAVg</td>
</tr>
<tr>
<td>LARGE</td>
<td>METHOD, NUMCycles</td>
</tr>
<tr>
<td>LAYOUT</td>
<td>MID, NUMENTries</td>
</tr>
<tr>
<td>LDIR</td>
<td>MID2, NUMHORZ</td>
</tr>
<tr>
<td>LEFT</td>
<td>MIL, NUMVERT</td>
</tr>
<tr>
<td>LESSEqual</td>
<td>MIN, POWER</td>
</tr>
<tr>
<td>LESSLimit</td>
<td>MINimum, POWERFactor</td>
</tr>
<tr>
<td>LESSThan</td>
<td>MINMax, POWERFactor</td>
</tr>
<tr>
<td>LETTER</td>
<td>MISO, OCCURS</td>
</tr>
<tr>
<td>LEVEL</td>
<td>MISOMOSI, ODD</td>
</tr>
<tr>
<td>LF</td>
<td>MIXed, ODD</td>
</tr>
<tr>
<td>LFRej</td>
<td>MIXED2, OFF</td>
</tr>
<tr>
<td>LIMITS</td>
<td>MKDir, ONCE</td>
</tr>
<tr>
<td>LIMIT</td>
<td>MM100BY150, ONCE</td>
</tr>
<tr>
<td>LIN</td>
<td>MM54BY86, ONFAIL</td>
</tr>
<tr>
<td>LINE</td>
<td>MODE, OPTION</td>
</tr>
<tr>
<td>LINEar</td>
<td>MODULATIONanalysis, OPTION</td>
</tr>
<tr>
<td>LINEFREQUENCY</td>
<td>MODE, OUT</td>
</tr>
<tr>
<td>LINEPeriod</td>
<td>MODEL, OUTrange</td>
</tr>
<tr>
<td>LINear</td>
<td>MODulation, OVERLoad</td>
</tr>
<tr>
<td>LIST</td>
<td>MOREEqual, OVERLoad</td>
</tr>
<tr>
<td>L3</td>
<td>MORElimit, OWNER</td>
</tr>
<tr>
<td>LOCK</td>
<td>MOREThan, PACKET</td>
</tr>
<tr>
<td>LOG</td>
<td>MOREthan, PAL</td>
</tr>
<tr>
<td>LOGIC</td>
<td>MOSI, PAPERSize</td>
</tr>
<tr>
<td>LOOP</td>
<td>MSB, PROPELAY</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Reserved Word</td>
<td>Reserved Word</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>PT_Fmt</td>
<td>PT_Fmt</td>
</tr>
<tr>
<td>PT_Order</td>
<td>PT_Order</td>
</tr>
<tr>
<td>PT_Off</td>
<td>PT_Off</td>
</tr>
<tr>
<td>PULSEWIDTH</td>
<td>PULSEWIDTH</td>
</tr>
<tr>
<td>PULSEWidth</td>
<td>PULSEWidth</td>
</tr>
<tr>
<td>PULSE</td>
<td>PULSE</td>
</tr>
<tr>
<td>PULSE</td>
<td>PULSE</td>
</tr>
<tr>
<td>PWIdth</td>
<td>PWIdth</td>
</tr>
<tr>
<td>QTY</td>
<td>QTY</td>
</tr>
<tr>
<td>QUALifier</td>
<td>QUALifier</td>
</tr>
<tr>
<td>QUALity</td>
<td>QUALity</td>
</tr>
<tr>
<td>RADIUS</td>
<td>RADIUS</td>
</tr>
<tr>
<td>RATE10K</td>
<td>RATE10K</td>
</tr>
<tr>
<td>RATE15K</td>
<td>RATE15K</td>
</tr>
<tr>
<td>RATE1K</td>
<td>RATE1K</td>
</tr>
<tr>
<td>RATE20K</td>
<td>RATE20K</td>
</tr>
<tr>
<td>RATE25K</td>
<td>RATE25K</td>
</tr>
<tr>
<td>RATE30K</td>
<td>RATE30K</td>
</tr>
<tr>
<td>RATE32K</td>
<td>RATE32K</td>
</tr>
<tr>
<td>RATE33K</td>
<td>RATE33K</td>
</tr>
<tr>
<td>RATE35K</td>
<td>RATE35K</td>
</tr>
<tr>
<td>RATE37K</td>
<td>RATE37K</td>
</tr>
<tr>
<td>RATE50K</td>
<td>RATE50K</td>
</tr>
<tr>
<td>RATE62K</td>
<td>RATE62K</td>
</tr>
<tr>
<td>RATE800K</td>
<td>RATE800K</td>
</tr>
<tr>
<td>RATIO</td>
<td>RATIO</td>
</tr>
<tr>
<td>RDELta</td>
<td>RDELta</td>
</tr>
<tr>
<td>RDSon</td>
<td>RDSon</td>
</tr>
<tr>
<td>REACTpwr</td>
<td>REACTpwr</td>
</tr>
<tr>
<td>READ</td>
<td>READ</td>
</tr>
<tr>
<td>READFile</td>
<td>READFile</td>
</tr>
<tr>
<td>READOUT</td>
<td>READOUT</td>
</tr>
<tr>
<td>RECALL</td>
<td>RECALL</td>
</tr>
<tr>
<td>RECORDSkew</td>
<td>RECORDSkew</td>
</tr>
<tr>
<td>RECORDlength</td>
<td>RECORDlength</td>
</tr>
<tr>
<td>RECTangular</td>
<td>RECTangular</td>
</tr>
<tr>
<td>REF</td>
<td>REF</td>
</tr>
<tr>
<td>REF1</td>
<td>REF1</td>
</tr>
<tr>
<td>REF2</td>
<td>REF2</td>
</tr>
<tr>
<td>REF3</td>
<td>REF3</td>
</tr>
<tr>
<td>REF4</td>
<td>REF4</td>
</tr>
<tr>
<td>REFLevel</td>
<td>REFLevel</td>
</tr>
<tr>
<td>REM</td>
<td>REM</td>
</tr>
<tr>
<td>REMote</td>
<td>REMote</td>
</tr>
<tr>
<td>Rename</td>
<td>Rename</td>
</tr>
<tr>
<td>REPEATstart</td>
<td>REPEATstart</td>
</tr>
<tr>
<td>RESERVED</td>
<td>RESERVED</td>
</tr>
<tr>
<td>RESET</td>
<td>RESET</td>
</tr>
<tr>
<td>RESOLUTION</td>
<td>RESOLUTION</td>
</tr>
<tr>
<td>RESptime</td>
<td>RESptime</td>
</tr>
<tr>
<td>RESUME</td>
<td>RESUME</td>
</tr>
<tr>
<td>RESULT</td>
<td>RESULT</td>
</tr>
<tr>
<td>Resistance</td>
<td>Resistance</td>
</tr>
<tr>
<td>Results</td>
<td>Results</td>
</tr>
</tbody>
</table>

**Appendix B: Reserved Words**
<table>
<thead>
<tr>
<th>Reserved Word</th>
<th>Reserved Word</th>
<th>Reserved Word</th>
<th>Reserved Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB</td>
<td>VERTAUTOset</td>
<td>WFMOutpre</td>
<td>XY</td>
</tr>
<tr>
<td>USBDevice</td>
<td>VERTDEFault</td>
<td>WHEN</td>
<td>XZero</td>
</tr>
<tr>
<td>USBTMC</td>
<td>VERTical</td>
<td>WIDth</td>
<td>Y</td>
</tr>
<tr>
<td>USBTmc</td>
<td>VIDEO</td>
<td>WIndow</td>
<td>YDElta</td>
</tr>
<tr>
<td>USE</td>
<td>VOLTAGESource</td>
<td>WORD</td>
<td>YES</td>
</tr>
<tr>
<td>USER</td>
<td>VOLTage</td>
<td>WORDSel</td>
<td>YMAX</td>
</tr>
<tr>
<td>V1X</td>
<td>VOLts</td>
<td>WRITE</td>
<td>YMIN</td>
</tr>
<tr>
<td>V2X</td>
<td>VRMS</td>
<td>WRITEFile</td>
<td>YMULT</td>
</tr>
<tr>
<td>VALUE</td>
<td>WAKEup</td>
<td>X</td>
<td>YOFF</td>
</tr>
<tr>
<td>VBArS</td>
<td>WAVEFORMS</td>
<td>XDELta</td>
<td>YT</td>
</tr>
<tr>
<td>VCEsat</td>
<td>WAVEform</td>
<td>XFF</td>
<td>YNUnits</td>
</tr>
<tr>
<td>VCRESTfactor</td>
<td>WAVEFrm</td>
<td>XINcr</td>
<td>YNUnits</td>
</tr>
<tr>
<td>VDELTa</td>
<td>WEIghting</td>
<td>XMAX</td>
<td>YNUnits</td>
</tr>
<tr>
<td>VENDORID</td>
<td>WFId</td>
<td>XMIN</td>
<td>YZErO</td>
</tr>
<tr>
<td>VERBose</td>
<td>WFMInpre</td>
<td>XNUnit</td>
<td>ZOOM</td>
</tr>
</tbody>
</table>

MSO4000 and DPO4000 Series Programmer Manual B-5
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>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQ:MAG</td>
<td>0</td>
</tr>
<tr>
<td>ACQ:MOD</td>
<td>SAM</td>
</tr>
<tr>
<td>ACQ:NUMAV</td>
<td>16</td>
</tr>
<tr>
<td>ACQ:NUME</td>
<td>INFI</td>
</tr>
<tr>
<td>ACQ:STATE</td>
<td>1</td>
</tr>
<tr>
<td>ACQ:STOPA</td>
<td>RUNST</td>
</tr>
<tr>
<td>ALI</td>
<td>0</td>
</tr>
<tr>
<td>AUX:PRO:FORCEDR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>AUX:PRO:GAIN</td>
<td>1.0000</td>
</tr>
<tr>
<td>BUS:B1:CAN:BITR</td>
<td>500000</td>
</tr>
<tr>
<td>BUS:B1:CAN:PRO</td>
<td>CANH</td>
</tr>
<tr>
<td>BUS:B1:CAN:SAMPLE</td>
<td>50</td>
</tr>
<tr>
<td>BUS:B1:CAN:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B1:DIS:FORM</td>
<td>HEX</td>
</tr>
<tr>
<td>BUS:B1:DIS:TYP</td>
<td>BUS</td>
</tr>
<tr>
<td>BUS:B1:I2C:ADDR:RWINC</td>
<td>0</td>
</tr>
<tr>
<td>BUS:B1:I2C:CLOC:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B1:I2C:DAT:SOU</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B1:LAB</td>
<td>&quot;Parallel&quot;</td>
</tr>
<tr>
<td>BUS:B1:LIN:BITR</td>
<td>19200</td>
</tr>
<tr>
<td>BUS:B1:LIN:IDFOR</td>
<td>NOPAR</td>
</tr>
<tr>
<td>BUS:B1:LIN:POL</td>
<td>NORM</td>
</tr>
<tr>
<td>BUS:B1:LIN:SAMPLE</td>
<td>50</td>
</tr>
<tr>
<td>BUS:B1:LIN:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B1:LIN:STAND</td>
<td>V2X</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT0:SOU</td>
<td>D0</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT10:SOU</td>
<td>D10</td>
</tr>
</tbody>
</table>
## Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS:B1:PAR:BIT11:SOU</td>
<td>D11</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT12:SOU</td>
<td>D12</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT14:SOU</td>
<td>D14</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT15:SOU</td>
<td>D15</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT16:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT17:SOU</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT18:SOU</td>
<td>CH3</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT19:SOU</td>
<td>CH4</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT1:SOU</td>
<td>D1</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT2:SOU</td>
<td>D2</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT3:SOU</td>
<td>D3</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT4:SOU</td>
<td>D4</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT5:SOU</td>
<td>D5</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT6:SOU</td>
<td>D6</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT7:SOU</td>
<td>D7</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT8:SOU</td>
<td>D8</td>
</tr>
<tr>
<td>BUS:B1:PAR:BIT9:SOU</td>
<td>D9</td>
</tr>
<tr>
<td>BUS:B1:PAR:CLOC:EDGE</td>
<td>RIS</td>
</tr>
<tr>
<td>BUS:B1:PAR:CLOC:ISCLOCK</td>
<td>NO</td>
</tr>
<tr>
<td>BUS:B1:PAR:CLOC:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B1:PAR:WID</td>
<td>16</td>
</tr>
<tr>
<td>BUS:B1:POS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:B1:RS232C:BITR</td>
<td>9600</td>
</tr>
<tr>
<td>BUS:B1:RS232C:DATAB</td>
<td>8</td>
</tr>
<tr>
<td>BUS:B1:RS232C:DELIM</td>
<td>LF</td>
</tr>
<tr>
<td>BUS:B1:RS232C:DIS</td>
<td>FRA</td>
</tr>
<tr>
<td>BUS:B1:RS232C:PAR</td>
<td>NON</td>
</tr>
<tr>
<td>BUS:B1:RS232C:POL</td>
<td>NORM</td>
</tr>
<tr>
<td>BUS:B1:RS232C:RX:SOU</td>
<td>OFF</td>
</tr>
<tr>
<td>BUS:B1:RS232C:TX:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B1:SPI:BIT0:MSB</td>
<td>MSB</td>
</tr>
<tr>
<td>BUS:B1:SPI:CLOC:POL</td>
<td>RIS</td>
</tr>
<tr>
<td>BUS:B1:SPI:CLOC:SOU</td>
<td>CH1</td>
</tr>
</tbody>
</table>
## Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS:B1:SPI:DAT:SIZ</td>
<td>8</td>
</tr>
<tr>
<td>BUS:B1:SPI:FRAMING</td>
<td>SS</td>
</tr>
<tr>
<td>BUS:B1:SPI:IDLET</td>
<td>5.0000E-6</td>
</tr>
<tr>
<td>BUS:B1:SPI:SEL:POL</td>
<td>LOW</td>
</tr>
<tr>
<td>BUS:B1:SPI:SEL:SOU</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B1:STATE</td>
<td>0</td>
</tr>
<tr>
<td>BUS:B1:TYP</td>
<td>PAR</td>
</tr>
<tr>
<td>BUS:B1:USB:BITR</td>
<td>FULL</td>
</tr>
<tr>
<td>BUS:B1:USB:PRO</td>
<td>DIFF</td>
</tr>
<tr>
<td>BUS:B1:USB:SOU:DIFF</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B1:USB:SOU:DMIN</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B1:USB:SOU:DPLU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:CAN:BITR</td>
<td>500000</td>
</tr>
<tr>
<td>BUS:B2:CAN:PRO</td>
<td>CANH</td>
</tr>
<tr>
<td>BUS:B2:CAN:SAMPLE</td>
<td>50</td>
</tr>
<tr>
<td>BUS:B2:CAN:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:DIS:FORM</td>
<td>HEX</td>
</tr>
<tr>
<td>BUS:B2:DIS:TYP</td>
<td>BUS</td>
</tr>
<tr>
<td>BUS:B2:I2C:ADDR:RWINC</td>
<td>0</td>
</tr>
<tr>
<td>BUS:B2:I2C:CLOC:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:LIN:STAND</td>
<td>&quot;Parallel&quot;</td>
</tr>
<tr>
<td>BUS:B2:LIN:BITR</td>
<td>19200</td>
</tr>
<tr>
<td>BUS:B2:LIN:IDFOR</td>
<td>NOPAR</td>
</tr>
<tr>
<td>BUS:B2:LIN:POL</td>
<td>NORM</td>
</tr>
<tr>
<td>BUS:B2:LIN:SAMPLE</td>
<td>50</td>
</tr>
<tr>
<td>BUS:B2:LIN:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:LIN:STAND</td>
<td>V2X</td>
</tr>
<tr>
<td>BUS:B2:PAR:BIT0:SOU</td>
<td>D0</td>
</tr>
<tr>
<td>BUS:B2:PAR:BIT14:SOU</td>
<td>D14</td>
</tr>
<tr>
<td>BUS:B2:PAR:BIT15:SOU</td>
<td>D15</td>
</tr>
<tr>
<td>BUS:B2:PAR:BIT16:SOU</td>
<td>CH1</td>
</tr>
</tbody>
</table>
## Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS:B2:PAR:BIT17:SOU</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B2:PAR:BIT18:SOU</td>
<td>CH3</td>
</tr>
<tr>
<td>BUS:B2:PAR:BIT19:SOU</td>
<td>CH4</td>
</tr>
<tr>
<td>BUS:B2:PAR:BIT1:SOU</td>
<td>D1</td>
</tr>
<tr>
<td>BUS:B2:PAR:BIT2:SOU</td>
<td>D2</td>
</tr>
<tr>
<td>BUS:B2:PAR:BIT4:SOU</td>
<td>D4</td>
</tr>
<tr>
<td>BUS:B2:PAR:BIT7:SOU</td>
<td>D7</td>
</tr>
<tr>
<td>BUS:B2:PAR:BIT8:SOU</td>
<td>D8</td>
</tr>
<tr>
<td>BUS:B2:PAR:CLOC:EDGE</td>
<td>RIS</td>
</tr>
<tr>
<td>BUS:B2:PAR:CLOC:ISCLOCK</td>
<td>NO</td>
</tr>
<tr>
<td>BUS:B2:PAR:CLOC:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:PAR:WID</td>
<td>16</td>
</tr>
<tr>
<td>BUS:B2:POS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:B2:RS232C:BITR</td>
<td>9600</td>
</tr>
<tr>
<td>BUS:B2:RS232C:DATAB</td>
<td>8</td>
</tr>
<tr>
<td>BUS:B2:RS232C:DELIM</td>
<td>LF</td>
</tr>
<tr>
<td>BUS:B2:RS232C:DIS</td>
<td>FRA</td>
</tr>
<tr>
<td>BUS:B2:RS232C:PAR</td>
<td>NON</td>
</tr>
<tr>
<td>BUS:B2:RS232C:POL</td>
<td>NORM</td>
</tr>
<tr>
<td>BUS:B2:RS232C:RX:SOU</td>
<td>OFF</td>
</tr>
<tr>
<td>BUS:B2:RS232C:TX:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:SPI:BITO</td>
<td>MSB</td>
</tr>
<tr>
<td>BUS:B2:SPI:CLOC:POL</td>
<td>RIS</td>
</tr>
<tr>
<td>BUS:B2:SPI:CLOC:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:SPI:FRAMING</td>
<td>SS</td>
</tr>
<tr>
<td>BUS:B2:SPI:IDLET</td>
<td>5.0000E-6</td>
</tr>
<tr>
<td>BUS:B2:SPI:SEL:POL</td>
<td>LOW</td>
</tr>
<tr>
<td>BUS:B2:SPI:SEL:SOU</td>
<td>CH2</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS:B2:STATE</td>
<td>0</td>
</tr>
<tr>
<td>BUS:B2:TYPO</td>
<td>PAR</td>
</tr>
<tr>
<td>BUS:B2:USB:BITR</td>
<td>FULL</td>
</tr>
<tr>
<td>BUS:B2:USB:PRO</td>
<td>DIFF</td>
</tr>
<tr>
<td>BUS:B2:USB:SOU:DIFF</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B2:USB:SOU:DMIN</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B2:USB:SOU:DPLU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B3:CAN:BITR</td>
<td>500000</td>
</tr>
<tr>
<td>BUS:B3:CAN:PRO</td>
<td>CANH</td>
</tr>
<tr>
<td>BUS:B3:CAN:SAMPLE</td>
<td>50</td>
</tr>
<tr>
<td>BUS:B3:CAN:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B3:DIS:FORM</td>
<td>HEX</td>
</tr>
<tr>
<td>BUS:B3:DIS:TYP</td>
<td>BUS</td>
</tr>
<tr>
<td>BUS:B3:I2C:ADDR:RWINC</td>
<td>0</td>
</tr>
<tr>
<td>BUS:B3:I2C:CLOC:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B3:I2C:DAT:SOU</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B3:LIN:BITR</td>
<td>19200</td>
</tr>
<tr>
<td>BUS:B3:LIN:IDFOR</td>
<td>NOPAR</td>
</tr>
<tr>
<td>BUS:B3:LIN:POL</td>
<td>NORM</td>
</tr>
<tr>
<td>BUS:B3:LIN:SAMPLE</td>
<td>50</td>
</tr>
<tr>
<td>BUS:B3:LIN:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B3:LIN:STAND</td>
<td>V2X</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT0:SOU</td>
<td>D0</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT10:SOU</td>
<td>D10</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT11:SOU</td>
<td>D11</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT12:SOU</td>
<td>D12</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT14:SOU</td>
<td>D14</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT15:SOU</td>
<td>D15</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT16:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT17:SOU</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT18:SOU</td>
<td>CH3</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT19:SOU</td>
<td>CH4</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT1:SOU</td>
<td>D1</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT2:SOU</td>
<td>D2</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT3:SOU</td>
<td>D3</td>
</tr>
</tbody>
</table>
Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS:B3:PAR:BIT4:SOUR</td>
<td>D4</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT5:SOUR</td>
<td>D5</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT6:SOUR</td>
<td>D6</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT7:SOUR</td>
<td>D7</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT8:SOUR</td>
<td>D8</td>
</tr>
<tr>
<td>BUS:B3:PAR:BIT9:SOUR</td>
<td>D9</td>
</tr>
<tr>
<td>BUS:B3:PAR:CLOC:EDGE</td>
<td>RIS</td>
</tr>
<tr>
<td>BUS:B3:PAR:CLOC:ISCLK</td>
<td>NO</td>
</tr>
<tr>
<td>BUS:B3:PAR:CLOC:SOUR</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B3:PAR:WID</td>
<td>16</td>
</tr>
<tr>
<td>BUS:B3:POS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:B3:RS232C:BITR</td>
<td>9600</td>
</tr>
<tr>
<td>BUS:B3:RS232C:DATAB</td>
<td>8</td>
</tr>
<tr>
<td>BUS:B3:RS232C:DELIM</td>
<td>LF</td>
</tr>
<tr>
<td>BUS:B3:RS232C:DIS</td>
<td>FRA</td>
</tr>
<tr>
<td>BUS:B3:RS232C:PAR</td>
<td>NON</td>
</tr>
<tr>
<td>BUS:B3:RS232C:POL</td>
<td>NORM</td>
</tr>
<tr>
<td>BUS:B3:RS232C:RX:SOUR</td>
<td>OFF</td>
</tr>
<tr>
<td>BUS:B3:RS232C:TX:SOUR</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B3:SPI:BITO</td>
<td>MSB</td>
</tr>
<tr>
<td>BUS:B3:SPI:CLOC:POL</td>
<td>RIS</td>
</tr>
<tr>
<td>BUS:B3:SPI:CLOC:SOUR</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B3:SPI:DAT:IN:POL</td>
<td>HIGH</td>
</tr>
<tr>
<td>BUS:B3:SPI:DAT:IN:SOUR</td>
<td>OFF</td>
</tr>
<tr>
<td>BUS:B3:SPI:DAT:SIZ</td>
<td>8</td>
</tr>
<tr>
<td>BUS:B3:SPI:FRAMING</td>
<td>SS</td>
</tr>
<tr>
<td>BUS:B3:SPI:IDLET</td>
<td>5.0000E-6</td>
</tr>
<tr>
<td>BUS:B3:SPI:SEL:POL</td>
<td>LOW</td>
</tr>
<tr>
<td>BUS:B3:SPI:SEL:SOUR</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B3:STATE</td>
<td>0</td>
</tr>
<tr>
<td>BUS:B3:TYP</td>
<td>PAR</td>
</tr>
<tr>
<td>BUS:B3:USB:BITR</td>
<td>FULL</td>
</tr>
<tr>
<td>BUS:B3:USB:PRO</td>
<td>DIFF</td>
</tr>
<tr>
<td>BUS:B3:USB:SOU:DIFF</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B3:USB:SOU:DMIN</td>
<td>CH2</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS:B3:USB:SOU:DPLU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B4:CAN:BITR</td>
<td>500000</td>
</tr>
<tr>
<td>BUS:B4:CAN:PRO</td>
<td>CANH</td>
</tr>
<tr>
<td>BUS:B4:CAN:SAMPLE</td>
<td>50</td>
</tr>
<tr>
<td>BUS:B4:CAN:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B4:DIS:FORM</td>
<td>HEX</td>
</tr>
<tr>
<td>BUS:B4:DIS:TYP</td>
<td>BUS</td>
</tr>
<tr>
<td>BUS:B4:I2C:ADDR:RWINC</td>
<td>0</td>
</tr>
<tr>
<td>BUS:B4:I2C:CLOC:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B4:I2C:DAT:SOU</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B4:LAB</td>
<td>&quot;Parallel&quot;</td>
</tr>
<tr>
<td>BUS:B4:LIN:BITR</td>
<td>19200</td>
</tr>
<tr>
<td>BUS:B4:LIN:IDFOR</td>
<td>NOPAR</td>
</tr>
<tr>
<td>BUS:B4:LIN:POL</td>
<td>NORM</td>
</tr>
<tr>
<td>BUS:B4:LIN:SAMPLE</td>
<td>50</td>
</tr>
<tr>
<td>BUS:B4:LIN:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B4:LIN:STAND</td>
<td>V2X</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT0:SOU</td>
<td>D0</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT10:SOU</td>
<td>D10</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT11:SOU</td>
<td>D11</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT12:SOU</td>
<td>D12</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT14:SOU</td>
<td>D14</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT15:SOU</td>
<td>D15</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT16:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT17:SOU</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT18:SOU</td>
<td>CH3</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT19:SOU</td>
<td>CH4</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT1:SOU</td>
<td>D1</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT2:SOU</td>
<td>D2</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT3:SOU</td>
<td>D3</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT4:SOU</td>
<td>D4</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT5:SOU</td>
<td>D5</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT6:SOU</td>
<td>D6</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT7:SOU</td>
<td>D7</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT8:SOU</td>
<td>D8</td>
</tr>
<tr>
<td>BUS:B4:PAR:BIT9:SOU</td>
<td>D9</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS:B4:PAR:CLOC:EDGE</td>
<td>RIS</td>
</tr>
<tr>
<td>BUS:B4:PAR:CLOC:ISCLOCK</td>
<td>NO</td>
</tr>
<tr>
<td>BUS:B4:PAR:CLOC:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B4:PAR:WID</td>
<td>16</td>
</tr>
<tr>
<td>BUS:B4:POS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:B4:RS232C:BITR</td>
<td>9600</td>
</tr>
<tr>
<td>BUS:B4:RS232C:DATAB</td>
<td>8</td>
</tr>
<tr>
<td>BUS:B4:RS232C:DELIM</td>
<td>LF</td>
</tr>
<tr>
<td>BUS:B4:RS232C:DIS</td>
<td>FRA</td>
</tr>
<tr>
<td>BUS:B4:RS232C:PAR</td>
<td>NON</td>
</tr>
<tr>
<td>BUS:B4:RS232C:POL</td>
<td>NORM</td>
</tr>
<tr>
<td>BUS:B4:RS232C:RX:SOU</td>
<td>OFF</td>
</tr>
<tr>
<td>BUS:B4:RS232C:TX:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B4:SPI:BITO</td>
<td>MSB</td>
</tr>
<tr>
<td>BUS:B4:SPI:CLOC:POL</td>
<td>RIS</td>
</tr>
<tr>
<td>BUS:B4:SPI:CLOC:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B4:SPI:DAT:IN:POL</td>
<td>HIGH</td>
</tr>
<tr>
<td>BUS:B4:SPI:DAT:IN:SOU</td>
<td>OFF</td>
</tr>
<tr>
<td>BUS:B4:SPI:DAT:SIZ</td>
<td>8</td>
</tr>
<tr>
<td>BUS:B4:SPI:FRAMING</td>
<td>SS</td>
</tr>
<tr>
<td>BUS:B4:SPI:IDLET</td>
<td>5.0000E-6</td>
</tr>
<tr>
<td>BUS:B4:SPI:SEL:POL</td>
<td>LOW</td>
</tr>
<tr>
<td>BUS:B4:SPI:SEL:SOU</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B4:STATE</td>
<td>0</td>
</tr>
<tr>
<td>BUS:B4:TYP</td>
<td>PAR</td>
</tr>
<tr>
<td>BUS:B4:USB:BITR</td>
<td>FULL</td>
</tr>
<tr>
<td>BUS:B4:USB:PRO</td>
<td>DIFF</td>
</tr>
<tr>
<td>BUS:B4:USB:SOU:DIFF</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:B4:USB:SOU:DMIN</td>
<td>CH2</td>
</tr>
<tr>
<td>BUS:B4:USB:SOU:DPLU</td>
<td>CH1</td>
</tr>
<tr>
<td>BUS:LOW:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:LOW:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:LOW:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:LOW:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>BUS:UPP:CH1</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>Command</td>
<td>Default value</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>BUS:UPP:CH2</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>BUS:UPP:CH3</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>BUS:UPP:CH4</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>CH1:AMPSVIAVOLT:ENA</td>
<td>0</td>
</tr>
<tr>
<td>CH1:AMPSVIAVOLT:FAC</td>
<td>10.0000</td>
</tr>
<tr>
<td>CH1:BAN</td>
<td>500.0000E+6</td>
</tr>
<tr>
<td>CH1:COUP</td>
<td>DC</td>
</tr>
<tr>
<td>CH1:DESK</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH1:INV</td>
<td>0</td>
</tr>
<tr>
<td>CH1:LAB</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>CH1:OFFS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH1:POS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH1:PRO:FORCEDR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH1:PRO:GAIN</td>
<td>1.0000</td>
</tr>
<tr>
<td>CH1:PRO:MOD</td>
<td>&quot;Other&quot;</td>
</tr>
<tr>
<td>CH1:PRO:PROPDEL</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH1:SCA</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>CH1:TER</td>
<td>1.0000E+6</td>
</tr>
<tr>
<td>CH1:YUN</td>
<td>&quot;/&quot;</td>
</tr>
<tr>
<td>CH2:AMPSVIAVOLT:ENA</td>
<td>0</td>
</tr>
<tr>
<td>CH2:AMPSVIAVOLT:FAC</td>
<td>10.0000</td>
</tr>
<tr>
<td>CH2:BAN</td>
<td>500.0000E+6</td>
</tr>
<tr>
<td>CH2:COUP</td>
<td>DC</td>
</tr>
<tr>
<td>CH2:DESK</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH2:INV</td>
<td>0</td>
</tr>
<tr>
<td>CH2:LAB</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>CH2:OFFS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH2:POS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH2:PRO:FORCEDR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH2:PRO:GAIN</td>
<td>1.0000</td>
</tr>
<tr>
<td>CH2:PRO:MOD</td>
<td>&quot;Other&quot;</td>
</tr>
<tr>
<td>CH2:PRO:PROPDEL</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH2:SCA</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>CH2:TER</td>
<td>1.0000E+6</td>
</tr>
<tr>
<td>CH2:YUN</td>
<td>&quot;/&quot;</td>
</tr>
<tr>
<td>CH3:AMPSVIAVOLT:ENA</td>
<td>0</td>
</tr>
<tr>
<td>CH3:AMPSVIAVOLT:FAC</td>
<td>10.0000</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH3:BAN</td>
<td>500.0000E+6</td>
</tr>
<tr>
<td>CH3:COUP</td>
<td>DC</td>
</tr>
<tr>
<td>CH3:DESK</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH3:INV</td>
<td>0</td>
</tr>
<tr>
<td>CH3:LAB</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>CH3:OFFS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH3:POS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH3:PRO:FORCEDR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH3:PRO:GAIN</td>
<td>1.0000</td>
</tr>
<tr>
<td>CH3:PRO:MOD</td>
<td>&quot;Other&quot;</td>
</tr>
<tr>
<td>CH3:PRO:PROPDEL</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH3:SCA</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>CH3:TER</td>
<td>1.0000E+6</td>
</tr>
<tr>
<td>CH3:YUN</td>
<td>&quot;V&quot;</td>
</tr>
<tr>
<td>CH4:AMPSVIAVOLT:ENA</td>
<td>0</td>
</tr>
<tr>
<td>CH4:AMPSVIAVOLT:FAC</td>
<td>10.0000</td>
</tr>
<tr>
<td>CH4:BAN</td>
<td>500.0000E+6</td>
</tr>
<tr>
<td>CH4:COUP</td>
<td>DC</td>
</tr>
<tr>
<td>CH4:DESK</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH4:INV</td>
<td>0</td>
</tr>
<tr>
<td>CH4:LAB</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>CH4:OFFS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH4:POS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH4:PRO:FORCEDR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH4:PRO:GAIN</td>
<td>1.0000</td>
</tr>
<tr>
<td>CH4:PRO:MOD</td>
<td>&quot;Other&quot;</td>
</tr>
<tr>
<td>CH4:PRO:PROPDEL</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CH4:SCA</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>CH4:TER</td>
<td>1.0000E+6</td>
</tr>
<tr>
<td>CH4:YUN</td>
<td>&quot;V&quot;</td>
</tr>
<tr>
<td>CURS:FUNC</td>
<td>OFF</td>
</tr>
<tr>
<td>CURS:HBA:POSITION1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CURS:HBA:POSITION2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CURS:HBA:UNI</td>
<td>BAS</td>
</tr>
<tr>
<td>CURS:MOD</td>
<td>IND</td>
</tr>
<tr>
<td>CURS:VBA:POSITION1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CURS:VBA:POSITION2</td>
<td>0.0E+0</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURS:VBA:UNI</td>
<td>SEC</td>
</tr>
<tr>
<td>CURS:XY:READOUT</td>
<td>RECT</td>
</tr>
<tr>
<td>CURS:XY:RECT:X:POSITION1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CURS:XY:RECT:X:POSITION2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CURS:XY:RECT:Y:POSITION1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>CURS:XY:RECT:Y:POSITION2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>D0:LAB</td>
<td>==</td>
</tr>
<tr>
<td>D0:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D0:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D10:LAB</td>
<td>==</td>
</tr>
<tr>
<td>D10:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D10:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D11:LAB</td>
<td>==</td>
</tr>
<tr>
<td>D11:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D11:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D12:LAB</td>
<td>==</td>
</tr>
<tr>
<td>D12:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D12:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D13:LAB</td>
<td>==</td>
</tr>
<tr>
<td>D13:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D13:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D14:LAB</td>
<td>==</td>
</tr>
<tr>
<td>D14:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D14:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D15:LAB</td>
<td>==</td>
</tr>
<tr>
<td>D15:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D15:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D1:L1AB</td>
<td>==</td>
</tr>
<tr>
<td>D1:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D1:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D2:LAB</td>
<td>==</td>
</tr>
<tr>
<td>D2:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D2:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D3:LAB</td>
<td>==</td>
</tr>
<tr>
<td>D3:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D3:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D4:LAB</td>
<td>==</td>
</tr>
</tbody>
</table>
Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>D4:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D4:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D5:LAB</td>
<td>**</td>
</tr>
<tr>
<td>D5:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D5:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D6:LAB</td>
<td>**</td>
</tr>
<tr>
<td>D6:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D6:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D7:LAB</td>
<td>**</td>
</tr>
<tr>
<td>D7:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D7:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D8:LAB</td>
<td>**</td>
</tr>
<tr>
<td>D8:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D8:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>D9:LAB</td>
<td>**</td>
</tr>
<tr>
<td>D9:POS</td>
<td>60.0000E-3</td>
</tr>
<tr>
<td>D9:THRE</td>
<td>1.4000</td>
</tr>
<tr>
<td>DAT:DEST</td>
<td>REF1</td>
</tr>
<tr>
<td>DAT:ENC</td>
<td>RIB</td>
</tr>
<tr>
<td>DAT:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>DAT:STAR</td>
<td>1</td>
</tr>
<tr>
<td>DAT:STOP</td>
<td>10000</td>
</tr>
<tr>
<td>DES:DISP</td>
<td>1</td>
</tr>
<tr>
<td>DIS:CLOC</td>
<td>1</td>
</tr>
<tr>
<td>DIS:HEI</td>
<td>MED</td>
</tr>
<tr>
<td>DIS:FORM</td>
<td>YT</td>
</tr>
<tr>
<td>DIS:GRA</td>
<td>FUL</td>
</tr>
<tr>
<td>DIS:INTENSIT:BACKL</td>
<td>HIGH</td>
</tr>
<tr>
<td>DIS:INTENSIT:GRA</td>
<td>75</td>
</tr>
<tr>
<td>DIS:INTENSIT:WAVE</td>
<td>35</td>
</tr>
<tr>
<td>DIS:PERS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>DIS:STY:DOT</td>
<td>0</td>
</tr>
<tr>
<td>DIS:TRIGF</td>
<td>0</td>
</tr>
<tr>
<td>HARDC:INKS</td>
<td>1</td>
</tr>
<tr>
<td>HARDC:LAY</td>
<td>LAN</td>
</tr>
<tr>
<td>HEAD</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIS:BOXP</td>
<td>20.0000, 20.0000, 80.0000, 80.0000</td>
</tr>
<tr>
<td>HIS:DIS</td>
<td>LINEA</td>
</tr>
<tr>
<td>HIS:MOD</td>
<td>OFF</td>
</tr>
<tr>
<td>HIS:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>HOR:DEL:MOD</td>
<td>1</td>
</tr>
<tr>
<td>HOR:DEL:TIM</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>HOR:POS</td>
<td>50.0000</td>
</tr>
<tr>
<td>HOR:RECO</td>
<td>10000</td>
</tr>
<tr>
<td>HOR:SCA</td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>LOC</td>
<td>NON</td>
</tr>
<tr>
<td>MATH:DEF</td>
<td>&quot;CH1+CH2&quot;</td>
</tr>
<tr>
<td>MATH:HOR:POS</td>
<td>50.0000</td>
</tr>
<tr>
<td>MATH:HOR:SCA</td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>MATH:HOR:UNI</td>
<td>&quot;s&quot;</td>
</tr>
<tr>
<td>MATH:LAB</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>MATH:SPECT:MAG</td>
<td>DB</td>
</tr>
<tr>
<td>MATH:SPECT:WIN</td>
<td>HAN</td>
</tr>
<tr>
<td>MATH:TYP</td>
<td>DUAL</td>
</tr>
<tr>
<td>MATH:VERT:POS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>MATH:VERT:SCA</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>MATH:VERT:UNI</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>MEASU:GAT</td>
<td>SCRE</td>
</tr>
<tr>
<td>MEASU:IMM:DEL:DIR</td>
<td>FORW</td>
</tr>
<tr>
<td>MEASU:IMM:DEL:EDGE1</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:IMM:DEL:EDGE2</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:IMM:SOU1</td>
<td>CH1</td>
</tr>
<tr>
<td>MEASU:IMM:SOU2</td>
<td>CH2</td>
</tr>
<tr>
<td>MEASU:IMM:TYP</td>
<td>PERI</td>
</tr>
<tr>
<td>MEASU:INDIC:STATE</td>
<td>OFF</td>
</tr>
<tr>
<td>MEASU:MEAS1:DEL:DIR</td>
<td>FORW</td>
</tr>
<tr>
<td>MEASU:MEAS1:DEL:EDGE1</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS1:DEL:EDGE2</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS1:SOU1</td>
<td>CH1</td>
</tr>
<tr>
<td>MEASU:MEAS1:SOU2</td>
<td>CH2</td>
</tr>
<tr>
<td>MEASU:MEAS1:STATE</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASU:MEAS1:TYP</td>
<td>PERI</td>
</tr>
<tr>
<td>MEASU:MEAS2:DEL:DIR</td>
<td>FORW</td>
</tr>
<tr>
<td>MEASU:MEAS2:DEL:EDGE1</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS2:DEL:EDGE2</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS2:SOU1</td>
<td>CH1</td>
</tr>
<tr>
<td>MEASU:MEAS2:SOU2</td>
<td>CH2</td>
</tr>
<tr>
<td>MEASU:MEAS2:STATE</td>
<td>0</td>
</tr>
<tr>
<td>MEASU:MEAS2:TYP</td>
<td>PERI</td>
</tr>
<tr>
<td>MEASU:MEAS3:DEL:DIR</td>
<td>FORW</td>
</tr>
<tr>
<td>MEASU:MEAS3:DEL:EDGE1</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS3:DEL:EDGE2</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS3:SOU1</td>
<td>CH1</td>
</tr>
<tr>
<td>MEASU:MEAS3:SOU2</td>
<td>CH2</td>
</tr>
<tr>
<td>MEASU:MEAS3:STATE</td>
<td>0</td>
</tr>
<tr>
<td>MEASU:MEAS3:TYP</td>
<td>PERI</td>
</tr>
<tr>
<td>MEASU:MEAS4:DEL:DIR</td>
<td>FORW</td>
</tr>
<tr>
<td>MEASU:MEAS4:DEL:EDGE1</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS4:DEL:EDGE2</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS4:SOU1</td>
<td>CH1</td>
</tr>
<tr>
<td>MEASU:MEAS4:SOU2</td>
<td>CH2</td>
</tr>
<tr>
<td>MEASU:MEAS4:STATE</td>
<td>0</td>
</tr>
<tr>
<td>MEASU:MEAS4:TYP</td>
<td>PERI</td>
</tr>
<tr>
<td>MEASU:MEAS5:DEL:DIR</td>
<td>FORW</td>
</tr>
<tr>
<td>MEASU:MEAS5:DEL:EDGE1</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS5:DEL:EDGE2</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS5:SOU1</td>
<td>CH1</td>
</tr>
<tr>
<td>MEASU:MEAS5:SOU2</td>
<td>CH2</td>
</tr>
<tr>
<td>MEASU:MEAS5:STATE</td>
<td>0</td>
</tr>
<tr>
<td>MEASU:MEAS5:TYP</td>
<td>PERI</td>
</tr>
<tr>
<td>MEASU:MEAS6:DEL:DIR</td>
<td>FORW</td>
</tr>
<tr>
<td>MEASU:MEAS6:DEL:EDGE1</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS6:DEL:EDGE2</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS6:SOU1</td>
<td>CH1</td>
</tr>
<tr>
<td>MEASU:MEAS6:SOU2</td>
<td>CH2</td>
</tr>
<tr>
<td>MEASU:MEAS6:STATE</td>
<td>0</td>
</tr>
<tr>
<td>MEASU:MEAS6:TYP</td>
<td>PERI</td>
</tr>
<tr>
<td>MEASU:MEAS7:DEL:DIR</td>
<td>FORW</td>
</tr>
</tbody>
</table>
Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASU:MEAS7:DEL:EDGE1</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS7:DEL:EDGE2</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS7:SOU1</td>
<td>CH1</td>
</tr>
<tr>
<td>MEASU:MEAS7:SOU2</td>
<td>CH2</td>
</tr>
<tr>
<td>MEASU:MEAS7:STATE</td>
<td>0</td>
</tr>
<tr>
<td>MEASU:MEAS7:TYP</td>
<td>PERI</td>
</tr>
<tr>
<td>MEASU:MEAS8:DEL:DIR</td>
<td>FORW</td>
</tr>
<tr>
<td>MEASU:MEAS8:DEL:EDGE1</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS8:DEL:EDGE2</td>
<td>RIS</td>
</tr>
<tr>
<td>MEASU:MEAS8:SOU1</td>
<td>CH1</td>
</tr>
<tr>
<td>MEASU:MEAS8:SOU2</td>
<td>CH2</td>
</tr>
<tr>
<td>MEASU:MEAS8:STATE</td>
<td>0</td>
</tr>
<tr>
<td>MEASU:MEAS8:TYP</td>
<td>PERI</td>
</tr>
<tr>
<td>MEASU:METH</td>
<td>A</td>
</tr>
<tr>
<td>MEASU:REFL:ABS:HIGH</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>MEASU:REFL:ABS:LOW</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>MEASU:REFL:ABS:MID1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>MEASU:REFL:ABS:MID2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>MEASU:REFL:METH</td>
<td>PERC</td>
</tr>
<tr>
<td>MEASU:REFL:PERC:HIGH</td>
<td>90.0000</td>
</tr>
<tr>
<td>MEASU:REFL:PERC:LOW</td>
<td>10.0000</td>
</tr>
<tr>
<td>MEASU:REFL:PERC:MID1</td>
<td>50.0000</td>
</tr>
<tr>
<td>MEASU:REFL:PERC:MID2</td>
<td>50.0000</td>
</tr>
<tr>
<td>MEASU:STATI:MODE</td>
<td>ALL</td>
</tr>
<tr>
<td>MEASU:STATI:WEI</td>
<td>32</td>
</tr>
<tr>
<td>MESS:BOX</td>
<td>271,82,271,98</td>
</tr>
<tr>
<td>MESS:SHOW</td>
<td>==</td>
</tr>
<tr>
<td>MESS:STATE</td>
<td>0</td>
</tr>
<tr>
<td>PICTB:DATEP</td>
<td>DEFLT</td>
</tr>
<tr>
<td>PICTB:IDP</td>
<td>OFF</td>
</tr>
<tr>
<td>PICTB:IMAGES</td>
<td>DEFLT</td>
</tr>
<tr>
<td>PICTB:PAPERS</td>
<td>DEFLT</td>
</tr>
<tr>
<td>PICTB:PAPERT</td>
<td>DEFLT</td>
</tr>
<tr>
<td>PICTB:PRINTQ</td>
<td>DEFLT</td>
</tr>
<tr>
<td>REF1:HOR:DEL:TIM</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>REF1:HOR:SCA</td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>REF1:VERT:POS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>Command</td>
<td>Default value</td>
</tr>
<tr>
<td>---------</td>
<td>---------------</td>
</tr>
<tr>
<td>REF1:VERT:SCA</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>REF2:HOR:DEL:TIM</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>REF2:HOR:SCA</td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>REF2:VERT:POS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>REF2:VERT:SCA</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>REF3:HOR:DEL:TIM</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>REF3:HOR:SCA</td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>REF3:VERT:POS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>REF3:VERT:SCA</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>REF4:HOR:DEL:TIM</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>REF4:HOR:SCA</td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>REF4:VERT:POS</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>REF4:VERT:SCA</td>
<td>100.0000E-3</td>
</tr>
<tr>
<td>SAV:ASSI:TYP</td>
<td>IMAG</td>
</tr>
<tr>
<td>SAV:IMAG:FILEF</td>
<td>PNG</td>
</tr>
<tr>
<td>SAV:IMAG:INKS</td>
<td>1</td>
</tr>
<tr>
<td>SAV:IMAG:LAY</td>
<td>LAN</td>
</tr>
<tr>
<td>SAV:WAVE:FILEF</td>
<td>SPREADS</td>
</tr>
<tr>
<td>SAV:WAVE:GATI</td>
<td>NON</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:STATE</td>
<td>0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B1:12C:COND</td>
<td>STAR</td>
</tr>
</tbody>
</table>
## Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B1:PAR:VAL</td>
<td>&quot;XXXXXXXXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>Command</td>
<td>Default value</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------------------------------------</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B3:CAN:COND</td>
<td>SOF</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B3:LIN:ERRTYPE</td>
<td>SYNC</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B3:SPI:COND</td>
<td>SS</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B3:USB:COND</td>
<td>SYNC</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B3:USB:ERRTYPE</td>
<td>PID</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B3:USB:HANDSHAKE</td>
<td>ANY</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B3:USB:QUAL</td>
<td>EQU</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B3:USB:SOFFRAMENUM</td>
<td>&quot;XXXXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B3:USB:SPECIALT</td>
<td>ANY</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B3:USB:TOKEN</td>
<td>ANY</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B4:CAN:COND</td>
<td>SOF</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B4:CAN:ID:VAL</td>
<td>&quot;XXXXXXXX&quot;</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B4:PAR:VAL</td>
<td>&quot;XXXXXXXXXXXXXXX&quot;</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B4:SPI:COND</td>
<td>SS</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B4:USB:COND</td>
<td>SYNC</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B4:USB:ERRTYPE</td>
<td>PID</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B4:USB:HANDSHAKET</td>
<td>ANY</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B4:USB:QUAL</td>
<td>EQU</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B4:USB:SOF FRAMENUM</td>
<td>&quot;XXXXXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B4:USB:SPECIALT</td>
<td>ANY</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:B4:USB:TOKENT</td>
<td>ANY</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:BUS:SOU</td>
<td>B1</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:EDGE:SLO</td>
<td>RIS</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:EDGE:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LEV</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LEV:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LEV:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LEV:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LEV:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LEV:MATH</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LEV:REF1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LEV:REF2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LEV:REF3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LEV:REF4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:FUNC</td>
<td>AND</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:INP:CH1</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:INP:D0</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:INP:D1</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:INP:D2</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:INP:D8</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:PA\text{T}:INP:CH1</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:PA\text{T}:INP:CH2</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:PA\text{T}:INP:CH3</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:PA\text{T}:INP:D0</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:PA\text{T}:INP:D1</td>
<td>X</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:PA\text{T}:INP:D10</td>
<td>X</td>
</tr>
</tbody>
</table>
Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:THR:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:THR:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:THR:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:THR:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:THR:MATH</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:THR:REF1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:THR:REF2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:THR:REF3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOGI:THR:REF4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOW:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOW:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOW:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOW:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOW:MATH</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOW:REF1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOW:REF2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOW:REF3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:LOW:REF4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:PULSEW:POL</td>
<td>POS</td>
</tr>
</tbody>
</table>
## Appendix C: Factory Defaults

### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:PULSEP:WHE</td>
<td>LESS</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:PULSEP:WID</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:RUNT:POL</td>
<td>POS</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:RUNT:WHE</td>
<td>OCCURS</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:RUNT:WID</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:SETH:CLOC:EDGE</td>
<td>RIS</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:SETH:CLOC:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:SETH:CLOC:THR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:SETH:SETT</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:SETH:THR:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:SETH:THR:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:SETH:THR:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:SETH:THR:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:SETH:THR:MATH</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:SETH:THR:REF1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:SETH:THR:REF2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:SETH:THR:REF3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:SETH:THR:REF4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:TRAN:POL</td>
<td>POS</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:TRAN:WHE</td>
<td>SLOW</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:TPEDG</td>
<td>EDG</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:UPP:CH1</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:UPP:CH2</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:UPP:CH3</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:UPP:CH4</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:UPP:MATH</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:UPP:REF1</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:UPP:REF2</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:UPP:REF3</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEARCH:SEARCH1:TRIG:A:UPP:REF4</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>SEL:BUS1</td>
<td>0</td>
</tr>
<tr>
<td>SEL:BUS2</td>
<td>0</td>
</tr>
<tr>
<td>SEL:BUS3</td>
<td>0</td>
</tr>
</tbody>
</table>
## Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEL:BUS4</td>
<td>0</td>
</tr>
<tr>
<td>SEL:CH1</td>
<td>1</td>
</tr>
<tr>
<td>SEL:CH2</td>
<td>0</td>
</tr>
<tr>
<td>SEL:CH3</td>
<td>0</td>
</tr>
<tr>
<td>SEL:CH4</td>
<td>0</td>
</tr>
<tr>
<td>SEL:CONTRO</td>
<td>CH1</td>
</tr>
<tr>
<td>SEL:D0</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D1</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D10</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D11</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D12</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D13</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D14</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D15</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D2</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D3</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D4</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D5</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D6</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D7</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D8</td>
<td>0</td>
</tr>
<tr>
<td>SEL:D9</td>
<td>0</td>
</tr>
<tr>
<td>SEL:MATH</td>
<td>0</td>
</tr>
<tr>
<td>SEL:REF1</td>
<td>0</td>
</tr>
<tr>
<td>SEL:REF2</td>
<td>0</td>
</tr>
<tr>
<td>SEL:REF3</td>
<td>0</td>
</tr>
<tr>
<td>SEL:REF4</td>
<td>0</td>
</tr>
<tr>
<td>SELECT:DALL</td>
<td>0</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:CAN:COND</td>
<td>SOF</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:CAN:DAT:STAR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:CAN:FRAME</td>
<td>DATA</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:CAN:ID:VAL</td>
<td>&quot;XXXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>Command</td>
<td>Default value</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:I2C:ADDR:VAL</td>
<td>&quot;XXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:I2C:COND</td>
<td>STAR</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:I2C:DAT:STAR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:LIN:COND</td>
<td>SYNCF</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:LIN:ERRTYPE</td>
<td>SYNC</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:LIN:ID:VAL</td>
<td>&quot;XXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:PAR:VAL</td>
<td>&quot;XXXXXXXXXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:SPI:COND</td>
<td>SS</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:SPI:DAT:STAR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:USB:COND</td>
<td>SYNC</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:USB:ENDP:VAL</td>
<td>&quot;XXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:USB:ERRTYPE</td>
<td>PID</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:USB:HANDSHAKET</td>
<td>ANY</td>
</tr>
</tbody>
</table>
## Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG:A:BUS:B1:USB:QUAL</td>
<td>EQU</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:USB:SOFFRAMENUM</td>
<td>&quot;XXXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:USB:SPECIALT</td>
<td>ANY</td>
</tr>
<tr>
<td>TRIG:A:BUS:B1:USB:TOCKET</td>
<td>ANY</td>
</tr>
<tr>
<td>TRIG:A:BUS:B2:CAN:DAT:STAR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:BUS:B2:I2C:DAT:STAR</td>
<td>0.0E+0</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG:A:BUS:B2:SPI:DAT:STAR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:CAN:COND</td>
<td>SOF</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:CAN:DAT:STAR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:CAN:FRAME</td>
<td>DATA</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:CAN:ID:MOD</td>
<td>ST</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:CAN:ID:VAL</td>
<td>&quot;XXXXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:I2C:ADDR:MOD</td>
<td>ADDR7</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:I2C:ADDR:TYP</td>
<td>USER</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:I2C:ADDR:VAL</td>
<td>&quot;XXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:I2C:COND</td>
<td>STAR</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:I2C:DAT:SIZ</td>
<td>1</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:I2C:DAT:STAR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:LIN:COND</td>
<td>SYNCF</td>
</tr>
</tbody>
</table>
## Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG:A:BUS:B3:LIN:ERTYPE</td>
<td>SYNC</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:LIN:ID:VAL</td>
<td>&quot;XXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:PAR:VAL</td>
<td>&quot;XXXXXXXXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:RS232C:COND</td>
<td>TXSTA</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:SPI:COND</td>
<td>SS</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:SPI:DAT:STAR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:USB:ADDR:HIVAL</td>
<td>&quot;XXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:USB:ADDR:VAL</td>
<td>&quot;XXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:USB:COND</td>
<td>SYNC</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:USB:ENDP:VAL</td>
<td>&quot;XXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:USB:ERTYPE</td>
<td>PID</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:USB:HANDSHAKET</td>
<td>ANY</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:USB:QUAL</td>
<td>EQU</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:USB:SOFRAMENUM</td>
<td>&quot;XXXXXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:USB:SPECIALT</td>
<td>ANY</td>
</tr>
<tr>
<td>TRIG:A:BUS:B3:USB:TOKENT</td>
<td>ANY</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:CAN:COND</td>
<td>SOF</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:CAN:DIR</td>
<td>NOCARE</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:CAN:QUAL</td>
<td>EQU</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:CAN:SIZ</td>
<td>1</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:CAN:STAR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:CAN:VAL</td>
<td>&quot;XXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:CAN:FRAME</td>
<td>DATA</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG:A:BUS:B4:CAN:ID:MOD</td>
<td>ST</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:CAN:ID:VAL</td>
<td>&quot;XXXXXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:I2C:ADDR:TYP</td>
<td>USER</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:I2C:ADDR:VAL</td>
<td>&quot;XXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:I2C:COND</td>
<td>STAR</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:I2C:DAT:STAR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:LIN:COND</td>
<td>SYNCF</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:LIN:ERRTYPE</td>
<td>SYNC</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:LIN:ID:VAL</td>
<td>&quot;XXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:PAR:VAL</td>
<td>&quot;XXXXXXXXXXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:RS232C:COND</td>
<td>TXSTA</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:SPI:COND</td>
<td>SS</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:SPI:DAT:STAR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:USB:ADDR:HIVAL</td>
<td>&quot;XXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:USB:ADDR:VAL</td>
<td>&quot;XXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:USB:COND</td>
<td>SYNC</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:USB:ENDP:VAL</td>
<td>&quot;XXX&quot;</td>
</tr>
<tr>
<td>Command</td>
<td>Default value</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:USB:ERRTYPE</td>
<td>PID</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:USB:HANDSHAKET</td>
<td>ANY</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:USB:QUAL</td>
<td>EQU</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:USB:SOFFRAMENUM</td>
<td>&quot;XXXXXXXXXXXX&quot;</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:USB:SPECIALT</td>
<td>ANY</td>
</tr>
<tr>
<td>TRIG:A:BUS:B4:USB:TOKENT</td>
<td>ANY</td>
</tr>
<tr>
<td>TRIG:A:BUS:SOU</td>
<td>B1</td>
</tr>
<tr>
<td>TRIG:A:EDGE:COUP</td>
<td>DC</td>
</tr>
<tr>
<td>TRIG:A:EDGE:SLO</td>
<td>RIS</td>
</tr>
<tr>
<td>TRIG:A:EDGE:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>TRIG:A:HOLD:TIM</td>
<td>20.0000E-9</td>
</tr>
<tr>
<td>TRIG:A:LEV</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:LEV:AUX</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:LEV:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:LEV:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:LEV:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:LEV:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:LEV:D0</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D1</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D10</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D11</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D12</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D13</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D14</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D15</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D2</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D3</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D4</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D5</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D6</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D7</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D8</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LEV:D9</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:CLA</td>
<td>LOGIC</td>
</tr>
<tr>
<td>TRIG:A:LOGI:FUNC</td>
<td>AND</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:CH1</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:CH2</td>
<td>X</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG:A:LOGI:INP.CH3</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP.CH4</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:CLOC:EDGE</td>
<td>RIS</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:CLOC:SOU</td>
<td>NON</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D0</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D1</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D10</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D11</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D12</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D13</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D14</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D15</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D2</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D3</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D4</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D5</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D6</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D7</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D8</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:INP:D9</td>
<td>X</td>
</tr>
<tr>
<td>TRIG:A:LOGI:PAT:WHE</td>
<td>TRU</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D0</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D1</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D10</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D11</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D12</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D13</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D14</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D15</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D2</td>
<td>1.4000</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG:A:LOGI:THR:D3</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D4</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D5</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D6</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D7</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D8</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOGI:THR:D9</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:LOW:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:LOW:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:LOW:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:LOW:D0</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D1</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D10</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D11</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D12</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D13</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D14</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D15</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D2</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D3</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D4</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D5</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D6</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D7</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D8</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:D9</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:LOW:EXT</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:MOD</td>
<td>AUTO</td>
</tr>
<tr>
<td>TRIG:A:PUL:CLA</td>
<td>WID</td>
</tr>
<tr>
<td>TRIG:A:PULSEW:POL</td>
<td>POS</td>
</tr>
<tr>
<td>TRIG:A:PULSEW:WHE</td>
<td>LESS</td>
</tr>
<tr>
<td>TRIG:A:PULSEW:WID</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIG:A:RUNT:POL</td>
<td>POS</td>
</tr>
<tr>
<td>TRIG:A:RUNT:WHE</td>
<td>OCCURS</td>
</tr>
<tr>
<td>TRIG:A:RUNT:WID</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIG:A:SETH:CLOC:EDGE</td>
<td>RIS</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG:A:SETH:CLOC:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>TRIG:A:SETH:CLOC:THR</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:SETH:DAT:SOU</td>
<td>NONE</td>
</tr>
<tr>
<td>TRIG:A:SETH:HOLDT</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIG:A:SETH:SETT</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D0</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D1</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D10</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D11</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D12</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D13</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D14</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D15</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D2</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D3</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D4</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D5</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D6</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D7</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D8</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:SETH:THR:D9</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:A:TRAN:DELT</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIG:A:TRAN:POL</td>
<td>POS</td>
</tr>
<tr>
<td>TRIG:A:TRAN:WHE</td>
<td>SLOW</td>
</tr>
<tr>
<td>TRIG:A:TYP</td>
<td>EDG</td>
</tr>
<tr>
<td>TRIG:A:UPP:CH1</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>TRIG:A:UPP:CH2</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>TRIG:A:UPP:CH3</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>TRIG:A:UPP:CH4</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>TRIG:A:VID:HOLD:FIELD</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:A:VID:LINE</td>
<td>1</td>
</tr>
<tr>
<td>TRIG:A:VID:POL</td>
<td>POS</td>
</tr>
<tr>
<td>Command</td>
<td>Default value</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>TRIG:A:VID:STAN</td>
<td>NTS</td>
</tr>
<tr>
<td>TRIG:A:VID:SYNC</td>
<td>ALLL</td>
</tr>
<tr>
<td>TRIG:B:BY</td>
<td>TIM</td>
</tr>
<tr>
<td>TRIG:B:EDGE:COUP</td>
<td>DC</td>
</tr>
<tr>
<td>TRIG:B:EDGE:SLO</td>
<td>RIS</td>
</tr>
<tr>
<td>TRIG:B:EDGE:SOU</td>
<td>CH1</td>
</tr>
<tr>
<td>TRIG:B:EVENTS:COUN</td>
<td>1</td>
</tr>
<tr>
<td>TRIG:B:LEV</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:B:LEV:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:B:LEV:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:B:LEV:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:B:LEV:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:B:LEV:D0</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D1</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D10</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D11</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D12</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D13</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D14</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D15</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D2</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D3</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D4</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D5</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D6</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D7</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D8</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LEV:D9</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:CH1</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:B:LOW:CH2</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:B:LOW:CH3</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:B:LOW:CH4</td>
<td>0.0E+0</td>
</tr>
<tr>
<td>TRIG:B:LOW:D0</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:D1</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:D10</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:D11</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:D12</td>
<td>1.4000</td>
</tr>
</tbody>
</table>
### Table C-1: Default values (cont.)

<table>
<thead>
<tr>
<th>Command</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG:B:LOW:D13</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:D14</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:D15</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:D2</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:D3</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:D4</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:D5</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:D6</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:D7</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:D8</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:LOW:D9</td>
<td>1.4000</td>
</tr>
<tr>
<td>TRIG:B:STATE</td>
<td>0</td>
</tr>
<tr>
<td>TRIG:B:TIM</td>
<td>8.0000E-9</td>
</tr>
<tr>
<td>TRIG:B:TYP</td>
<td>EDG</td>
</tr>
<tr>
<td>TRIG:B:UPP:CH1</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>TRIG:B:UPP:CH2</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>TRIG:B:UPP:CH3</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>TRIG:B:UPP:CH4</td>
<td>800.0000E-3</td>
</tr>
<tr>
<td>TRIG:EXT:PRO</td>
<td>1.0000</td>
</tr>
<tr>
<td>VERB</td>
<td>0</td>
</tr>
<tr>
<td>ZOO:MOD</td>
<td>0</td>
</tr>
<tr>
<td>ZOO:ZOOM1:HOR:POS</td>
<td>50.0000</td>
</tr>
<tr>
<td>ZOO:ZOOM1:HOR:SCA</td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>ZOO:ZOOM1:POS</td>
<td>50.0000</td>
</tr>
<tr>
<td>ZOO:ZOOM1:SCA</td>
<td>4.0000E-6</td>
</tr>
<tr>
<td>ZOO:ZOOM1:STATE</td>
<td>0</td>
</tr>
</tbody>
</table>
Appendix D: Programming Example

The following series of commands and queries illustrate many of the most common commands and techniques. In this example, data sent from the controller computer to the oscilloscope is prefaced with the > symbol. Replies from the oscilloscope have no preface.

> rem “Check for any messages, and clear them from the queue.”
> *esr?
128
> allev?
:ALLEV 401,”Power on; “

> rem “Set the scope to the default state.”
> factory

> rem “Set the scope parameters that differ from the defaults.”
> ch1:scale 2.0
> hor:m:scale 100e-6
> trig:a:level 2.4

> rem “Start a single sequence acquisition.”
> acquire:stopafter sequence
> acquire:state on
> rem “Wait for the acquisition to complete.”
> rem “Note: your controller program time-out must be set long enough to handle the wait.”
> *opc?
1
> rem “Use the oscilloscope built-in measurements to measure the waveform you acquired.”
> measu:immed:type mean
> measu:immed:value?
:MEASUREMENT:IMMED:VALUE 1.2767
> rem “Be sure to use the *esr? query to check for measurement errors.”
> measu:immed:type freq
> measu:immed:value
>:MEASUREMENT:IMMED:VALUE 9.9100E+37
> *esr?
  16
> allev?
>:ALLEV 2202,"Measurement error, No period found “

> rem “Query out the waveform points, for later analysis on your controller computer.”
> data:encdg ascii
> curve?
>:CURVE 7,6,5,5,5,6,6,6,8 […]

> rem “Query out the parameters used for calculating the times and voltages of the waveform points.”
> wfmpre?
>:WFMPRE:BYT_NR 1;BIT_NR 8;ENCDG ASCII;BN_FMT RI;BYT_OR MSB;NR_PT 10000; […]
Index

A
ACQuire:MAXSamplerate?, 2-72
ACQuire?, 2-71
ACQuire:MAgnivu, 2-71
ACQuire:MODE, 2-72
ACQuire:NUMACq?, 2-73
ACQuire:NUMAVg, 2-73
ACQuire:STATE, 2-74
ACQuire:STOPAfter, 2-75
Acquisition Command Group, 2-11
Alias Command Group, 2-12
ALLas, 2-75
ALLas[:STATE], 2-78
ALLas:CATalog?, 2-76
ALLas:DEFINE, 2-76
ALLas:DELETE, 2-77
ALLas:DELETE[:NAME], 2-78
ALLas:DELETE:ALL, 2-78
ALLEV?, 2-79
AUTOSet, 2-79
AUTOSet:ENABLE, 2-80
AUXin:PRObe:AUTOZero, 2-81
AUXin:PRObe:DEGAuss::STATE?, 2-82
AUXin:PRObe:FORCEDRange, 2-82
AUXin:PRObe:ID:SERnumber?, 2-83
AUXin:PRObe:ID:TYPE?, 2-83
AUXin:PRObe:RESistance?, 2-83
AUXin?, 2-80
AUXin:PRObe, 2-80
AUXin:PRObe:COMMAND, 2-81
AUXin:PRObe:DEGAuss, 2-82
AUXin:PRObe:GAIN, 2-83
AUXin:PRObe:SIGnal, 2-84
AUXin:PRObe:UNIts?, 2-84
BUS:B<x>:AUDio:DATa:SOUrce, 2-89
BUS:B<x>:AUDio:DISplay:FORMat, 2-89
BUS:B<x>:AUDio:FRAME:SIZE, 2-90
BUS:B<x>:AUDio:FRAMESync:POLarity, 2-90
BUS:B<x>:AUDio:FRAMESync:SOUrce, 2-91
BUS:B<x>:AUDio:TYPE, 2-91
BUS:B<x>:AUDio:WORDSel:POLarity, 2-92
BUS:B<x>:AUDio:WORDSel:SOUrce, 2-92
BUS:B<x>:CAN:BITRate, 2-93
BUS:B<x>:CAN:PRObe, 2-94
BUS:B<x>:CAN:SAMPLEpoint, 2-94
BUS:B<x>:CAN:SOUrce, 2-94
BUS:B<x>:DISplay:FORMat, 2-95
BUS:B<x>:DISplay:TYPe, 2-96
BUS:B<x>:FLEXray:BITRate, 2-96
BUS:B<x>:FLEXray:CHANnel, 2-97
BUS:B<x>:FLEXray:SIGnal, 2-97
BUS:B<x>:FLEXray:SOUrce, 2-98
BUS:B<x>:I2C:ADDRes:s:RWINclude, 2-98
BUS:B<x>:I2C::{CLOCK|SCLK}:SOUrce, 2-99
BUS:B<x>:I2C::{DATA|SDATA}:SOUrce, 2-99
BUS:B<x>:LIN:BITRate, 2-100
BUS:B<x>:LIN:IDFORmat, 2-100
BUS:B<x>:LIN:POLARity, 2-101
BUS:B<x>:LIN:SAMPLEpoint, 2-101
BUS:B<x>:LIN:SOUrce, 2-102
BUS:B<x>:LIN:STANDard, 2-102
BUS:B<x>:PARallel:BIT<x>:SOUrce, 2-103
BUS:B<x>:PARallel:CHANnel:EDGE, 2-103
BUS:B<x>:PARallel:CHANnel:ISCLOcked, 2-103
BUS:B<x>:PARallel:CHANnel:SOUrce, 2-104
BUS:B<x>:PARallel:WIDTh, 2-104
BUS:B<x>:POSition, 2-104
BUS:B<x>:RS232C:BITRate, 2-105
BUS:B<x>:RS232C:DATABits, 2-105
BUS:B<x>:RS232C:DELMiter, 2-106
BUS:B<x>:RS232C:DISplaymode, 2-106
BUS:B<x>:RS232C:PARity, 2-106
BUS:B<x>:RS232C:POLarity, 2-107
BUS:B<x>:RS232C:RX:SOUrce, 2-107
BUS:B<x>:RS232C:TX:SOUrce, 2-108
BUS:B<x>:SPI:BITOrder, 2-108
BUS:B<x>:SPI:DATA:SIZE, 2-111
BUS:B<x>:SPI:DATA::{IN|MISO}:POLARity, 2-109
BUS:B<x>:SPI:DATA{:IN|:MISO}:SOURce, 2-110
BUS:B<x>:SPI:DATA{:OUT|:MOSI}:POLARity, 2-110
BUS:B<x>:SPI:DATA{:OUT|:MOSI}:SOURce, 2-110
BUS:B<x>:SPI:FRAMING, 2-111
BUS:B<x>:SPI{{:CLOCK|:SCLK}:POLARity, 2-109
BUS:B<x>:SPI{{:CLOCK|:SCLK}:SOURce, 2-109
BUS:B<x>:SPI{{:SELect|:SS}:POLARity, 2-112
BUS:B<x>:SPI{{:SELect|:SS}:SOURce, 2-112
BUS:B<x>:USB:BITRate, 2-114
BUS:B<x>:USB:PRObe, 2-114
BUS:B<x>:USB:SOURce:DIFferential, 2-115
BUS:B<x>:USB:SOURce:DMINus, 2-115
BUS:B<x>:USB:SOURce:DPLUS, 2-115
BUS:B<x>:USB:SOURce:DIFFerential, 2-115
BUS:B<x>:USB:SOURce:DMINus, 2-115
BUS:B<x>:USB:SOURce:DPLUS, 2-115
BUS:LOWerthreshold:CH<x>, 2-116
BUS:THReshold:CH<x>, 2-117
BUS:THReshold:D<x>, 2-117
BUS:UPPerthreshold:CH<x>, 2-118
BUS?, 2-85
BUS:B<x>:LABel, 2-99
BUS:B<x>:STATE, 2-113
BUS:B<x>:TYPE, 2-113
BUSY?, 2-118

C
*CAL?, 2-119
CALibrate:FACTory:STATus?, 2-119
CALibrate:INTERnal:STARt, 2-120
CALibrate:INTERnal:STATus?, 2-120
CALibrate:RES ults:FACTory?, 2-121
CALibrate:RES ults:SPC?, 2-122
CALibrate:INTERnal, 2-120
CALibrate:RES ults?, 2-121
Calibration and Diagnostic Command Group, 2-17
CH<x>:AMPSVIAVOLTs:ENAble, 2-122
CH<x>:AMPSVIAVOLTs:FACTor, 2-123
CH<x>:PRObe:AUTOZero, 2-128
CH<x>:PRObe:COMMAND, 2-128
CH<x>:PRObe:DEGAUss, 2-129
CH<x>:PRObe:DEGAUss:STATE?, 2-129
CH<x>:PRObe:FORCEDRange, 2-129
CH<x>:PRObe:ID:SERnumber?, 2-131
CH<x>:PRObe:ID:TYPE?, 2-131
CH<x>:PRObe:MODel, 2-132
CH<x>:PRObe:PROPDELay, 2-132
CH<x>:PRObe:RECDESkew?, 2-132
CH<x>:PRObe:RESistance?, 2-133
CH<x>:PRObe:SIGnal, 2-133
CH<x>:PRObe:UNIts?, 2-133
CH<x>:TERmination, 2-135
CH<x>?, 2-122
CH<x>:BANwidth, 2-123
CH<x>:COUpling, 2-124
CH<x>:DESKew, 2-124
CH<x>:INVert, 2-125
CH<x>:LABel, 2-125
CH<x>:OFFSet, 2-126
CH<x>:POSition, 2-127
CH<x>:PRObe?, 2-128
CH<x>:PRObe:GAIN, 2-130
CH<x>:PRObe:ID?, 2-130
CH<x>:SCAle, 2-134
CH<x>:YUNits, 2-135
CLEARMenu, 2-136
*CLS, 2-136
Command Groups, 2-11
Cursor Command Group, 2-17
CURSor:HBAr:s:POSITION<x>, 2-138
CURSor:VBAr:s:ALTERNATE<x>?, 2-141
CURSor:VBAr:s:HPoS<x>?, 2-142
CURSor:VBAr:s:POSITION<x>, 2-142
CURSor:VBAr:s:VDELta?, 2-144
CURSor:XY:POLar:RADIUS:DELta?, 2-144
CURSor:XY:POLar:RADIUS:POSITION<x>?, 2-145
CURSor:XY:POLar:RADIUS:UNIts?, 2-145
CURSor:XY:POLar:THETA:DELta?, 2-145
CURSor:XY:POLar:THETA:POSITION<x>?, 2-145
CURSor:XY:POLar:THETA:UNIts?, 2-146
CURSor:XY:PRODUCT:DELta?, 2-146
CURSor:XY:PRODUCT:POSITION<x>?, 2-146
CURSor:XY:PRODUCT:UNIts?, 2-146
CURSor:XY:RATIO:DELta?, 2-147
CURSor:XY:RATIO:POSITION<x>?, 2-147
CURSor:XY:RATIO:UNIts?, 2-147
CURSor:XY:RECTangular:X:DELta?, 2-148
CURSor:XY:RECTangular:X:POSITION<x>?, 2-148
CURSor:XY:RECTangular:X:UNIts?, 2-148
CURSor:XY:RECTangular:Y:DELta?, 2-148
CURSor:XY:RECTangular:Y:POSITION<x>?, 2-148
CURSor:XY:RECTangular:Y:UNIts?, 2-149
CURSor?, 2-136
CURSor:DDT?, 2-137
CURSor:FUNCTION, 2-137
CURSor:HBArs?, 2-138
CURSor:HBArs:DELTa?, 2-138
CURSor:HBArs:UNIts, 2-139
CURSor:HBArs:USE, 2-139
CURSor:MODe, 2-140
CURSor:VBArs?, 2-140
CURSor:VBArs:DELTa?, 2-141
CURSor:VBArs:UNIts, 2-143
CURSor:VBArs:USE, 2-143
CURSor:XY:READOUT, 2-147
CURVe, 2-149

D
D<x>, 2-167
D<x>:LABel, 2-167
D<x>:POSition, 2-168
D<x>:THRESHold, 2-168
DATa, 2-151
DATa:DESprintation, 2-151
DATa:ENCdg, 2-152
DATa:SOUrce, 2-153
DATa:STARt, 2-154
DATa:STOP, 2-155
DATE, 2-156
*DDT, 2-156
*DESE, 2-157
*ESR?, 2-157
DESkew, 2-158
DESkew:DISPlay, 2-158
DIAg:LOOP:OPTion:NTIMes, 2-159
DIAg:SELe<function>, 2-161
DIAg:LOOP:OPTion, 2-158
DIAg:LOOP:STOP, 2-159
DIAg:RESULT:FLAg?, 2-160
DIAg:RESULT:LOG?, 2-160
DIAg:SELe<function>, 2-161
DIAg:STATE, 2-162
Display Command Group, 2-19
DISplay:DIGiltal:HELght, 2-163
DISplay:INTENSITy:BACKLight, 2-165
DISplay:INTENSITy:GRAticule, 2-165
DISplay:INTENSITy:WAVEform, 2-165
DISplay:STYLE:DOTsonly, 2-167
DISplay?, 2-162
DISplay:CLOCk, 2-162
DISplay:FORMat, 2-163
DISplay:GRAticule, 2-164
DISplay:INTENSITy?, 2-164
DISplay:PERS sistence, 2-166

E
*ES, 2-169
*ESR?, 2-169
Ethernet Command Group, 2-20
ETHERnet:DNS:IPADDress, 2-170
ETHERnet:ENET:ADDress?, 2-171
ETHERnet:GATEway:IPADDress, 2-171
ETHERnet:PING:STATUS?, 2-174
ETHERnet:DHCPbootp, 2-170
ETHERnet:DOMAINname, 2-171
ETHERnet:HTTPPort, 2-172
ETHERnet:IPADDress, 2-172
ETHERnet:NAME, 2-173
ETHERnet:PASSWord, 2-173
ETHERnet:PING, 2-173
ETHERnet:SUBNETMask, 2-174
EVENT?, 2-174
EVM<function>, 2-175
EV Qty?, 2-175

F
FACTory, 2-176
File System Command Group, 2-21
FILESystem:FREESpace?, 2-180
FILESystem:WRITEFile, 2-183
FILESystem?, 2-177
FILESystem:CWD, 2-177
FILESystem:DELETE, 2-178
FILESystem:DIR?, 2-179
FILESystem:FORMat, 2-179
FILESystem:LDIR?, 2-180
FILESystem:MKDir, 2-181
FILESystem:READFile, 2-181
FILESystem:RNAME, 2-182
FILESystem:RMDir, 2-182
FPANEL:PRESS, 2-183
FPANEL:TURN, 2-185

G
GPIBUsb:ADDress?, 2-186
GPIBUsb:ID?, 2-186
H
Hard copy Command Group, 2-22
HARDCopy, 2-186
HARDCopy:ACTIVeprinter, 2-187
HARDCopy:PRINTer:ADD, 2-189
HARDCopy:PRINTer:DELeTe, 2-189
HARDCopy:PRINTer:LIST?, 2-189
HARDCopy:PRINTer:REName, 2-189
HARDCopy:INKSaver, 2-187
HARDCopy:LAYout, 2-188
HARDCopy:PREVIEW, 2-188
HEADer, 2-190
Histogram Command Group, 2-24
HISTogram?, 2-191
HISTogram:BOX, 2-191
HISTogram:BOXPcnt, 2-192
HISTogram:COUNT, 2-193
HISTogram:DATa?, 2-193
HISTogram:DISPLAY, 2-194
HISTogram:END?, 2-195
HISTogram:MODE, 2-195
HISTogram:SOURCE, 2-196
HISTogram:START?, 2-197
Horizontal Command Group, 2-24
HORizontal:DELay:MODE, 2-198
HORizontal:DELay:TIME, 2-198
HORizontal:DIGital:RECORDlength:MAGrivu?, 2-199
HORizontal:DIGital:RECORDlength:MAin?, 2-199
HORizontal:DIGital:RECORDlength:MAInMAGivu?, 2-199
HORizontal:DIGital:SAMPLERate:MAGrivu?, 2-199
HORizontal:DIGital:SAMPLERate:MAIn?, 2-199
HORizontal:DIGital:SAMPLERate:REViewstate?, 2-200
HORizontal:RECORDlength, 2-200
HORizontal:SAMPLERate, 2-201
HORizontal?, 2-197
HORizontal:POSITION, 2-200
HORizontal:SCALE, 2-201

I
ID?, 2-202
*IDN?, 2-202

L
LANGuage, 2-202
LOck, 2-203
*LRN?, 2-203

M
Mark Command Group, 2-25
MARK, 2-205
MARK:SELECTed:FOCUS?, 2-206
MARK:SELECTed:MARKSINCOlumn?, 2-207
MARK:SELECTed:OWNER?, 2-207
MARK:SELECTed:SOUrce?, 2-207
MARK:SELECTed:STArt?, 2-207
MARK:SELECTed:STATe?, 2-208
MARK:SELECTed:ZOOm:POSition?, 2-208
MARK:CREATE, 2-205
MARK:DELeTe, 2-205
MARK:FREE?, 2-206
MARK:SELECTed:END?, 2-206
MARK:TOTAL?, 2-208
Math Command Group, 2-26
MATH[1]:HOrizontal:POSITION, 2-211
MATH[1]:HOrizontal:SCALE, 2-211
MATH[1]:HOrizontal:UNIts, 2-211
MATH[1]:SPECTral:MAG, 2-212
MATH[1]:SPECTral:WINdow, 2-212
MATH[1]:VERTical:POSITION, 2-213
MATH[1]:VERTical:SCALE, 2-214
MATH[1]:VERTical:UNIts, 2-214
MATH[1]?, 2-209
MATH[1]:DEFine, 2-209
MATH[1]:TYPE, 2-213
{MATH|MATH1}:LABel, 2-208
MATHVAR?, 2-215
MATHVAR:VAR<x>, 2-215
Measurement Command Group, 2-27
MEASUREment:CLEARSnapshot, 2-217
MEASUREment:IMMed:DELay:DIRection, 2-218
MEASUREment:IMMed:DELay:EDGE<x>, 2-219
MEASUREment:IMMed:DELay?, 2-218
MEASUREment:IMMed:SOUrce<x>, 2-219
MEASUREment:IMMed:TYPE, 2-220
MEASUREment:IMMed:UNIts?, 2-223
MEASUREment:IMMed:VALue?, 2-225
MEASUREment:INDICators:HORIZ<x>?, 2-226
MEASUREment:INDICators:NUMHORIZ?, 2-226
MEASUREment:INDICators:NUMVERT?, 2-227
MEASUREment:INDICators:STATE, 2-227
MEASUREment:INDICators:VERT<x>?, 2-228
MEASUREment:INDICators?, 2-226
MEASUREment:MEAS<x>:COUNt, 2-228
MEASUREment:MEAS<x>:DELay:DIREction, 2-229
MEASUrement:MEAS<x>:DELay:EDGE<x>, 2-230
MEASUrement:MEAS<x>:DELay?, 2-229
MEASUrement:MEAS<x>:MAXimum?, 2-230
MEASUrement:MEAS<x>:MEAN?, 2-231
MEASUrement:MEAS<x>:MINimum?, 2-231
MEASUrement:MEAS<x>:SOURCE<x>, 2-231
MEASUrement:MEAS<x>:STATE, 2-232
MEASUrement:MEAS<x>:STDEv?, 2-233
MEASUrement:MEAS<x>:TYPE, 2-233
MEASUrement:MEAS<x>:UNitS?, 2-236
MEASUrement:MEAS<x>:VALUE?, 2-238
MEASUrement:MEAS<x>?, 2-228
MEASUrement:REFLevel:ABSolute:HIGH, 2-240
MEASUrement:REFLevel:ABSolute:LOW, 2-240
MEASUrement:REFLevel:ABSolute:MID, 2-241
MEASUrement:REFLevel:ABSolute:MID<x>, 2-242
MEASUrement:REFLevel:MET hod, 2-243
MEASUrement:REFLevel:PERCent:HIGH, 2-243
MEASUrement:REFLevel:PERCent:LOW, 2-244
MEASUrement:REFLevel:PERCent:MID, 2-245
MEASUrement:REFLevel:PERCent:MID<x>, 2-246
MEASUrement:REFLevel:PERCent:MID2, 2-245
MEASUrement:REFLevel?, 2-239
MEASUrement:SNAPSHOT, 2-246
MEASUrement:STATIstics, 2-247
MEASUrement:STATIstics:MODE, 2-247
MEASUrement:STATIstics:WEIghting, 2-248
MEASUrement?, 2-216
MEASUrement:GAting, 2-217
MEASUrement:IMMed?, 2-217
MEASUrement:MET hod, 2-239
MESSage, 2-248
MESSage:BOX, 2-248
MESSage:CLEAR, 2-249
MESSage:SHOW, 2-250
MESSage:STATE, 2-252
Miscellaneous Command Group, 2-30

N
NEWpass, 2-253

O
*OPC, 2-253

P
PASSWord, 2-254
PICTBridge:DATEPrint, 2-254
PICTBridge:IMAGESize, 2-256
PICTBridge:PAPERSize, 2-257
PICTBridge:PAPERType, 2-258
PICTBridge:PRINTQual, 2-259
PICTBridge:DEFault, 2-255
PICTBridge:IDPrint, 2-255
POWer:HARMonics:DISPLAY:SELECT, 2-261
POWer:HARMonics:DISPLAY:TYPE, 2-262
POWer:HARMonics:FREQRef, 2-262
POWer:HARMonics:FREQRef:FIXEDFREQValue, 2-263
POWer:HARMonics:IEC:CLASS, 2-264
POWer:HARMonics:IEC:FILTER, 2-264
POWer:HARMonics:IEC:FUNDamental, 2-265
POWer:HARMonics:IEC:GROUPing, 2-265
POWer:HARMonics:IEC:INPUTPOWer, 2-266
POWer:HARMonics:IEC:LINEFREQuency, 2-266
POWer:HARMonics:IEC:OBSPERiod, 2-267
POWer:HARMonics:IEC:POWERFActor, 2-267
POWer:HARMonics:IEC:FUNDamental:
CALCmethod, 2-268
POWer:HARMonics:IEC:FUNDamental:USER:
CURRent, 2-268
POWer:HARMonics:IEC:LINEFREQuency, 2-269
POWer:HARMonics:IEC:POWERLEVEL, 2-269
POWer:HARMonics:NR_HARMonics, 2-270
POWer:HARMonics:RESults:HAR<1-400>:FREQuency?, 2-270
POWer:HARMonics:RESults:HAR<1-400>:IECCLASSLIMit?, 2-271
POWer:HARMonics:RESults:HAR<1-400>:LIMit?, 2-271
POWer:HARMonics:RESults:HAR<1-400>:PHASE?, 2-272
POWer:HARMonics:RESults:HAR<1-400>:RMS:
ABSolute?, 2-272
POWer:HARMonics:RESults:HAR<1-400>:RMS:
PERCent?, 2-273
POWer:HARMonics:RESults:HAR<1-400>:TEST:
IEC:CLASSTLIMAmit?, 2-273
POWer:HARMonics:RESults:HAR<1-400>:TEST:
IEC:NORMAL?, 2-274
POWer:HARMonics:RESults:HAR<1-400>:TEST:
IEC:POHCLIMit?, 2-274
POWer:SWLoss:REFLevel:PERCent: LOWCurrent, 2-317
POWer:SWLoss:REFLevel:PERCent: LOWVoltage, 2-318
POWer:SWLoss:TOFF:ENERGY:MAX?, 2-318
POWer:SWLoss:TOFF:ENERGY:MEAN?, 2-319
POWer:SWLoss:TOFF:ENERGY:MIN?, 2-319
POWer:SWLoss:TOFF:POWER:MAX?, 2-320
POWer:SWLoss:TOFF:POWER:MEAN?, 2-320
POWer:SWLoss:TOFF:POWER:MIN?, 2-320
POWer:SWLoss:TON:ENERGY:MEAN?, 2-321
POWer:SWLoss:TON:ENERGY:MIN?, 2-321
POWer:SWLoss:TON:POWER:MEAN?, 2-321
POWer:SWLoss:TON:POWER:MIN?, 2-321
POWer:SWLoss:TOTal:ENERGY:MAX?, 2-322
POWer:SWLoss:TOTal:ENERGY:MEAN?, 2-322
POWer:SWLoss:TOTal:ENERGY:MIN?, 2-322
POWer:SWLoss:TOTal:POWER:MAX?, 2-322
POWer:SWLoss:TOTal:POWER:MEAN?, 2-322
POWer:SWLoss:TOTal:POWER:MIN?, 2-322
POWer:CURRENTSOurce, 2-259
POWer:DISplay, 2-260
POWer:GATESOurce, 2-260
POWer:GATing, 2-261
POWer:INDICators, 2-280
POWer:QUALity:IRMS?, 2-289
POWer:QUALity:VRMS?, 2-291
POWer:RIPPle, 2-296
POWer:RIPPle:SOUrce, 2-299
POWer:SOA:LOG:XMAX, 2-301
POWer:SOA:LOG:XMIN, 2-302
POWer:SOA:LOG:YMAX, 2-303
POWer:SOA:LOG:YMIN, 2-303
POWer:SOA:PLOTTYpE, 2-307
POWer:STATIstics, 2-308
POWer:SWLoss:RDSon, 2-315
POWer:SWLoss:VCEsat, 2-325
POWer:TYPe, 2-326
POWer:VOLTAGESOurce, 2-327
*PSC, 2-327
*PUD, 2-328

R *
*RCL, 2-329
RECALL:SETUp, 2-329

S *
*SAV, 2-335
Save and Recall Command Group, 2-40
SAVe:EVENTable:BUS<x>, 2-336
SAVe:IMAGe:FILEFormat, 2-337
SAVe:WA VEform:FILEFormat, 2-340
SAVe:WA VEform:GATIng, 2-340
SAVe:ASSIgn:TYPe, 2-335
SAVe:IMAGe, 2-336
SAVe:IMAGe:INKSaver, 2-337
SAVe:IMAGe:LAYout, 2-338
SAVe:SETUp, 2-338
SAVe:WA VEform, 2-339
Search Commands Group, 2-41
SEARCH:SEARCH<x>:COPY, 2-342
SEARCH:SEARCH<x>:STATE, 2-343
SEARCH:SEARCH<x>:TOTAL?, 2-343
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:AUDio:DATa:QUALi fi er, 2-346
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:CONDition, 2-346
SEARCH: SEARCH<x>: TRIGGER:A: BUS:B<x>: SPI: DATA{::MISO|::IN}: VALUE, 2-372
SEARCH: SEARCH<x>: TRIGGER:A: BUS:B<x>: SPI: DATA{::MOSI|::OUT}: VALUE, 2-372
SEARCH: SEARCH<x>: TRIGGER:A: BUS:B<x>: USB: SOFRAMENUMBER, 2-381
SEARCH: SEARCH<x>: TRIGGER:A: BUS:B<x>: USB: SPECIAL TYPE, 2-382
SEARCH: SEARCH<x>: TRIGGER:A: BUS: SOURCE, 2-383
SEARCH: SEARCH<x>: TRIGGER:A: BUS?:, 2-343
SEARCH: SEARCH<x>: TRIGGER:A: EDGE: SLOPE, 2-384
SEARCH: SEARCH<x>: TRIGGER:A: EDGE: SOURCE, 2-384
SEARCH: SEARCH<x>: TRIGGER:A: LEVEL, 2-384
SEARCH: SEARCH<x>: TRIGGER:A: LEVEL: CH<x>, 2-385
SEARCH: SEARCH<x>: TRIGGER:A: LEVEL: MATH, 2-385
SEARCH: SEARCH<x>: TRIGGER:A: LEVEL: REF<x>, 2-385
SEARCH: SEARCH<x>: TRIGGER:A: LOGIC: FUNCTION, 2-386
SEARCH: SEARCH<x>: TRIGGER:A: LOGIC: INPUT: CH<x>, 2-386
SEARCH: SEARCH<x>: TRIGGER:A: LOGIC: INPUT: D<x>, 2-387
SEARCH: SEARCH<x>: TRIGGER:A: LOGIC: INPUT: MATH, 2-388
SEARCH: SEARCH<x>: TRIGGER:A: LOGIC: INPUT: REF<x>, 2-388
SEARCH: SEARCH<x>: TRIGGER:A: LOGIC: PATTERN: INPUT: D<x>, 2-389
SEARCH: SEARCH<x>: TRIGGER:A: LOGIC: PATTERN: WHEN, 2-390
SEARCH: SEARCH<x>: TRIGGER:A: LOGIC: THRESHOLD: CH<x>, 2-392
SEARCH: SEARCH<x>: TRIGGER:A: LOGIC: THRESHOLD: MATH, 2-392
SEARCH: SEARCH<x>: TRIGGER:A: LOGIC: THRESHOLD: MATH, 2-392
SEARCH: SEARCH<x>: TRIGGER:A: LOGIC: THRESHOLD: REF<x>, 2-392
SEARCH: SEARCH<x>: TRIGGER:A: LEVEL: LOW LIMIT: CH<x>, 2-393
SEARCH: SEARCH<x>: TRIGGER:A: LEVEL: LOW LIMIT: MATH, 2-393
SEARCH: SEARCH<x>: TRIGGER:A: LEVEL: LOW LIMIT: MATH, 2-393
SEARCH: SEARCH<x>: TRIGGER:A: LEVEL: LOW LIMIT: REF<x>, 2-393
SEARCH: SEARCH<x>: TRIGGER:A: PULSE WIDTH: POLARITY, 2-394
Index

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth: SOURCE, 2-394
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth: WHEN, 2-394
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth: WIDTH, 2-395
SEARCH:SEARCH<x>:TRIGger:A:RUNT: POLarity, 2-395
SEARCH:SEARCH<x>:TRIGger:A:RUNT: SOURCE, 2-396
SEARCH:SEARCH<x>:TRIGger:A:RUNT: WHEN, 2-396
SEARCH:SEARCH<x>:TRIGger:A:RUNT: WIDTH, 2-397
SEARCH:SEARCH<x>:TRIGger:A:SETHold: CLOCK:EDGE, 2-397
SEARCH:SEARCH<x>:TRIGger:A:SETHold: CLOCK:SOURce, 2-398
SEARCH:SEARCH<x>:TRIGger:A:SETHold: CLOCK:THRESHold, 2-398
SEARCH:SEARCH<x>:TRIGger:A:SETHold: DATA: SOURCE, 2-398
SEARCH:SEARCH<x>:TRIGger:A:SETHold: DATA: THRESHold, 2-399
SEARCH:SEARCH<x>:TRIGger:A:SETHold: HOLDTime, 2-400
SEARCH:SEARCH<x>:TRIGger:A:SETHold: SETTime, 2-400
SEARCH:SEARCH<x>:TRIGger:A:SETHold: REF<>, 2-401
SEARCH:SEARCH<x>:TRIGger:A:SETHold: THRESHold [:MATH][MATH1], 2-401
SEARCH:SEARCH<x>:TRIGger:A:TYPEc, 2-403
SEARCH:SEARCH<x>:TRIGger:A:UPPerrhreshold: CH<>, 2-404
SEARCH:SEARCH<x>:TRIGger:A:UPPerrhreshold: MATH, 2-404
SEARCH:SEARCH<x>:TRIGger:A:UPPerrhreshold: REF<>, 2-404
SEARCH:SEARCH<x>:TRIGger:A {:TRANSition}: RISEFall{:DELTAtime}, 2-401
SEARCH:SEARCH<x>:TRIGger:A {:TRANSition}: RISEFall{:POLarity}, 2-402
SEARCH:SEARCH<x>:TRIGger:A {:TRANSition}: RISEFall{:SOURCE}, 2-402
SEARCH:SEARCH<x>:TRIGger:A {:TRANSition}: RISEFall{:WHEn}, 2-403
SEARCH?, 2-341
SELECT, 2-405
SELECT:B<>, 2-405
SELECT:CH<>, 2-406
SELECT:CONTROL, 2-406
SELECT:D<>, 2-407
SELECT:MATH[1], 2-407
SELECT:REF<>, 2-408
SELECT?, 2-409
SETUP<>, 2-410
SETUP<>, 2-410
SETUP<>, 2-410
*ST?E, 2-411
Status and Error Command Group, 2-48
*STB?, 2-411

T

TEKSecure, 2-412
TIME, 2-413
TOTALuptime?, 2-413
*TRG, 2-414
Trigger Command Group, 2-49
TRIGger, 2-414
TRIGger:A:B<>:AUDio:CONDition, 2-416
TRIGger:A:B<>:AUDio:DATA: HIValue, 2-417
TRIGger:A:B<>:AUDio:DATA: OFFSet, 2-417
TRIGger:A:B<>:AUDio:DATA: QUALifier, 2-417
TRIGger:A:B<>:AUDio:DATA: VALue, 2-418
TRIGger:A:B<>:AUDio:DATA: WORD, 2-418
TRIGger:A:B<>:CAN:CONDition, 2-419
TRIGger:A:B<>:CAN:DATA: DIRECTION, 2-419
TRIGger:A:B<>:CAN:DATA: QUALifier, 2-420
TRIGger:A:B<>:CAN:DATA: SIZE, 2-421
TRIGger:A:B<>:CAN:DATA: VALue, 2-421
TRIGger:A:B<>:CAN:FRAMEtype, 2-422
TRIGger:A:B<>:CAN {:IDENTifier}: ADDRess: MODE, 2-422
TRIGger:A:B<>:CAN {:IDENTifier}: ADDRess: VALue, 2-423
TRIGger:A:B<>:FLEXray:CONDition, 2-423
TRIGger:A:B<>:FLEXray:CYCLEcount: HIValue, 2-424
TRIGger:A:BUS:B<x>:FLEXray:CYCLEcount:QUALifier, 2-424
TRIGger:A:BUS:B<x>:FLEXray:DATA:HAVALUE, 2-426
TRIGger:A:BUS:B<x>:FLEXray:DATA:OFFSET, 2-426
TRIGger:A:BUS:B<x>:FLEXray:DATA:QUALifier, 2-427
TRIGger:A:BUS:B<x>:FLEXray:DATA:SIZE, 2-427
TRIGger:A:BUS:B<x>:FLEXray:EOFTYPE, 2-428
TRIGger:A:BUS:B<x>:FLEXray:ERRTYPE, 2-429
TRIGger:A:BUS:B<x>:FLEXray:FRAMEID:HVALUE, 2-430
TRIGger:A:BUS:B<x>:FLEXray:FRAMEID:QUALifier, 2-430
TRIGger:A:BUS:B<x>:FLEXray:FRAMEID:VALUE, 2-431
TRIGger:A:BUS:B<x>:FLEXray:FRAMEID:FRAMEType, 2-431
TRIGger:A:BUS:B<x>:FLEXray:HEADER:CRC, 2-432
TRIGger:A:BUS:B<x>:FLEXray:HEADER:CYCLEcount, 2-432
TRIGger:A:BUS:B<x>:FLEXray:HEADER:FRAMEID, 2-433
TRIGger:A:BUS:B<x>:FLEXray:HEADER:INDBits, 2-433
TRIGger:A:BUS:B<x>:I2C:CONDITION, 2-436
TRIGger:A:BUS:B<x>:I2C:DATA:DIRECTION, 2-436
TRIGger:A:BUS:B<x>:I2C:DATA:SIZE, 2-437
TRIGger:A:BUS:B<x>:I2C:DATA:VALUE, 2-437
TRIGger:A:BUS:B<x>:I2C:LINE:CONDITION, 2-438
TRIGger:A:BUS:B<x>:I2C:DATA:HVALUE, 2-438
TRIGger:A:BUS:B<x>:I2C:DATA:SIZE, 2-440
TRIGger:A:BUS:B<x>:I2C:DATA:VALUE, 2-440
TRIGger:A:BUS:B<x>:I2C:DATA:ERRORTYPE, 2-440
TRIGger:A:BUS:B<x>:I2C:DATA:INPUT:CH<x>, 2-463
TRIGger:A:BUS:B<x>:PARALLEL:VALUE, 2-442
TRIGger:A:BUS:B<x>:RS232C:CONDITION, 2-442
TRIGger:A:BUS:B<x>:SPI:CONDITION, 2-444
TRIGger:A:BUS:B<x>:SPI:DATA:{IN|MISO}:VALUE, 2-445
TRIGger:A:BUS:B<x>:USB:ADDRESS:HVALUE, 2-446
TRIGger:A:BUS:B<x>:USB:ADDRESS:VALUE, 2-446
TRIGger:A:BUS:B<x>:USB:ADDRESS:WARNING, 2-447
TRIGger:A:BUS:B<x>:USB:DATA:HVALUE, 2-448
TRIGger:A:BUS:B<x>:USB:DATA:OFFSET, 2-448
TRIGger:A:BUS:B<x>:USB:DATA:SIZE, 2-449
TRIGger:A:BUS:B<x>:USB:DATA:TYPE, 2-450
TRIGger:A:BUS:B<x>:USB:DATA:VALUE, 2-450
TRIGger:A:BUS:B<x>:USB:ERRORTYPE, 2-451
TRIGger:A:BUS:B<x>:USB:HANDSHAKE:TYPE, 2-452
TRIGger:A:BUS:B<x>:USB:QUALIFIER, 2-453
TRIGger:A:BUS:B<x>:USB:WARNING, 2-453
TRIGger:A:BUS:B<x>:USB:SOFRAME:NUMBER, 2-454
TRIGger:A:BUS:B<x>:USB:SPECIAL:TYPE, 2-454
TRIGger:A:BUS:B<x>:USB:TOKEN:TYPE, 2-455
TRIGger:A:BUS:B<x>:USB:SOURCE, 2-456
TRIGger:A:EDGE:COUPLING, 2-456
TRIGger:A:EDGE:SLOPE, 2-457
TRIGger:A:EDGE:SLOPE:TIME, 2-459
TRIGger:A:LEVEL:AUXIN, 2-460
TRIGger:A:LEVEL:CH<x>, 2-460
TRIGger:A:LEVEL:D<x>, 2-461
TRIGger:A:LOGIC:CLASS, 2-461
TRIGger:A:LOGIC:FUNCTION, 2-462
TRIGger:A:LOGIC:INPUT:CH<x>, 2-463
TRIGger:A:LOGIc:INPut:D<x>, 2-465
TRIGger:A:LOGIc:INPut?, 2-466
TRIGger:A:LOGIc:INPut:CLOCk:SOUrce, 2-466
TRIGger:A:LOGIc:INPut:D<x>, 2-466
TRIGger:A:LOGIc:INPut:WHen, 2-466
TRIGger:A:LOGIc:INPut:WHen:
  LESSLimit, 2-467
TRIGger:A:LOGIc:INPut:WHen:
  MORELimit, 2-467
TRIGger:A:LOGIc:INPut?, 2-467
TRIGger:A:LOGIc:PATtern:DELTatime, 2-465
TRIGger:A:LOGIc:PATtern:INPut:D<x>, 2-466
TRIGger:A:LOGIc:PATtern:WHen, 2-466
TRIGger:A:LOGIc:PATtern?, 2-465
TRIGger:A:LOGIc:THReshold:CH<x>, 2-468
TRIGger:A:LOGIc:THReshold:D<x>, 2-469
TRIGger:A:LOWerthreshold:CH<x>, 2-469
TRIGger:A:LOWerthreshold:D<x>, 2-470
TRIGger:A:LOWerthreshold:{:EXT|:AUX}, 2-470
TRIGger:A:PULse:CLASS, 2-471
TRIGger:A:PULSEWidth:POLarity, 2-472
TRIGger:A:PULSEWidth:SOUrce, 2-473
TRIGger:A:PULSEWidth:WHen, 2-473
TRIGger:A:PULSEWidth:WIDth, 2-474
TRIGger:A:PULSEWidth?:, 2-474
TRIGger:A:RUNT:POLarity, 2-475
TRIGger:A:RUNT:SOUrce, 2-475
TRIGger:A:RUNT?:, 2-477
TRIGger:A:SETHold:CLOCk:EDGE, 2-478
TRIGger:A:SETHold:CLOCk:SOUrce, 2-478
TRIGger:A:SETHold:CLOCk:THReshold, 2-479
TRIGger:A:SETHold:CLOCk?:, 2-477
TRIGger:A:SETHold:DATa:SOUrce, 2-480
TRIGger:A:SETHold:DATa:THReshold, 2-481
TRIGger:A:SETHold:DATa?:, 2-479
TRIGger:A:SETHold:HOLDTime, 2-481
TRIGger:A:SETHold:SETTime, 2-482
TRIGger:A:SETHold:THReshold:CH<x>, 2-482
TRIGger:A:SETHold:THReshold:D<x>, 2-483
TRIGger:A:UPPerthreshold:CH<x>, 2-486
TRIGger:A:VIDeo:CUSTom:LINEPeriod, 2-488
TRIGger:A:VIDeo:CUSTom:SCAN, 2-488
TRIGger:A:VIDeo:CUSTom:SYNCInterval, 2-489
TRIGger:A:VIDeo:CUSTom {:FORMat|:TYPE}, 2-487
TRIGger:A:VIDeo:HDtv:FORMat, 2-490
TRIGger:A:VIDeo:HOLDoFF:FIELD, 2-490
TRIGger:A:VIDeo:LINE, 2-491
TRIGger:A:VIDeo:POLarity, 2-492
TRIGger:A:VIDeo:SOUrce, 2-492
TRIGger:A:VIDeo:STANdard, 2-493
TRIGger:A:VIDeo {:SYNC|:FIELD}, 2-493
TRIGger:A {:TRANsition:RISEFall}: DELTatime, 2-483
TRIGger:A {:TRANsition:RISEFall}:POLarity, 2-484
TRIGger:A {:TRANsition:RISEFall}:SOUrce, 2-484
TRIGger:A {:TRANsition:RISEFall}:WHen, 2-485
TRIGger:A {:TRANsition:RISEFall}?:, 2-483
TRIGger:B:EDGE:COUpling, 2-495
TRIGger:B:EDGE:SLOpe, 2-496
TRIGger:B:EDGE:SOUrce, 2-496
TRIGger:B:EVENTS:COUNt, 2-497
TRIGger:B:LEVel:CH<x>, 2-498
TRIGger:B:LEVel:D<x>, 2-499
TRIGger:B:LOWerthreshold:CH<x>, 2-499
TRIGger:B:LOWerthreshold:D<x>, 2-500
TRIGger:B:UPPerthreshold:CH<x>, 2-502
TRIGger:B:EXTernal:PRObe, 2-502
TRIGger:B:EXTernal:YUNIts?, 2-503
TRIGger:A, 2-414
TRIGger:A:BUS, 2-416
TRIGger:A:EDGE?, 2-456
TRIGger:A:HOLDoff?, 2-458
TRIGger:A:LEVEL, 2-459
TRIGger:A:LOGIc?, 2-461
TRIGger:A:MODE, 2-471
TRIGger:A:PULse?, 2-471
TRIGger:A:RUNT?, 2-474
TRIGger:A:RUNT:WHen, 2-476
TRIGger:A:SETHold?, 2-477
TRIGger:A:TYPe, 2-486
TRIGger:A:VIDeo?, 2-487
TRIGger:B, 2-494
TRIGger:B:BY, 2-494
TRIGger:B:EDGE?, 2-495
TRIGger:B:EVENTS?, 2-497
TRIGger:B:LEVEL, 2-498
TRIGger:B:STATE, 2-500
TRIGger:B:TIME, 2-501
TRIGger:B:TYPe, 2-501
TRIGger:B:EXTernal?, 2-502
TRIGger:B:FRQQuency?, 2-503
TRIGger:B:STATE?, 2-503
*TST?, 2-504
Index

U
UNLock, 2-504
USBDevice:CONFigure, 2-505
USBTMC:PRODUCTID:DECimal?, 2-505
USBTMC:PRODUCTID:HEXadecimal?, 2-506
USBTMC:SERIALnumber?, 2-506
USBTMC:VENDORID:DECimal?, 2-507
USBTMC:VENDORID:HEXadecimal?, 2-507
USBTMC?, 2-505

V
VERBose, 2-507
Vertical Command Group, 2-60

W
*WAI, 2-508
Waveform Transfer Command Group, 2-63
WAVFrm?, 2-509
WFMIpre?, 2-509
WFMIpre:BIT_Nr, 2-510
WFMIpre:BN_Fmt, 2-510
WFMIpre:BYT_Nr, 2-511
WFMIpre:BYT_Nr:BN_Fmt, 2-511
WFMIpre:ENCdg, 2-512
WFMIpre:NR_Pt, 2-512
WFMIpre:PT_Fmt, 2-513
WFMIpre:PT_Off, 2-514
WFMIpre:XINcr, 2-514
WFMIpre:XUNit, 2-515
WFMIpre:XZEro, 2-515
WFMIpre:YMUlt, 2-516
WFMIpre:YOFf, 2-517
WFMIpre:YNIt, 2-517
WFMOutpre:FRACtional?, 2-522
WFMOutpre?, 2-519
WFMOutpre:BIT_Nr, 2-519
WFMOutpre:BN_Fmt, 2-520
WFMOutpre:BYT_Nr, 2-520
WFMOutpre:BYT_Or, 2-521
WFMOutpre:ENCdg, 2-521
WFMOutpre:NR_Pt?, 2-522
WFMOutpre:PT_Fmt?, 2-523
WFMOutpre:PT_Off?, 2-523
WFMOutpre:PT_ORder?, 2-524
WFMOutpre:WFId?, 2-524
WFMOutpre:XINcr?, 2-525
WFMOutpre:YNIt, 2-525
WFMOutpre:XZEro?, 2-526
WFMOutpre:YMUlt?, 2-527
WFMOutpre:YOFf?, 2-527
WFMOutpre:YNIt?, 2-528
WFMOutpre:YZEro?, 2-528

Z
Zoom Command Group, 2-68
ZOOm:ZOOM<x>:FACTOR?, 2-530
ZOOm:ZOOM<x>:POSItion, 2-530
ZOOm:ZOOM<x>:SCAle, 2-530
ZOOm:ZOOM<x>:STATE, 2-531
ZOOm?, 2-528
ZOOm:MODE, 2-529
ZOOm:ZOOM<x>?, 2-529