TSG4100A Series
RF Signal Generators

Programmer Manual

www.tektronix.com
077-1079-00

Tektronix

Contacting Tektronix

Tektronix, Inc.
14150 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

Syntax and Commands

Status and Errors

Appendices

Preface

This programmer manual covers the TSG4100A Series RF Signal Generators. It provides information on operating your generator using LAN, the General Purpose Interface Bus (GPIB), and RS-232.

This manual is composed of the following sections

- *Getting Started* outlines how to use LAN, the GPIB interface, and RS-232.
- *Syntax and Commands* defines the syntax used in command descriptions, presents a list of all command subsystems, and presents detailed descriptions of all programming commands.
- *Status and Events* describes how the status and Events Reporting system operates and presents a list of all system errors.
- *Appendices* provides additional information including character charts, LAN specifications, GPIB interface specifications, RS-232 interface specifications, and factory initialization settings.

Related Documentation

The following documents provide operating information for the TSG4100A Series instruments. They are available on the Tektronix Web site at www.tektronix.com/manuals.

- *TSG4100A Series RF Signal Generator User Manual*
  This manual contains detailed operating, UI navigation, and other information to help you operate the instrument.
- *TSG4100A Series RF Signal Generator Specifications and Performance Verification Technical Reference*
  This technical reference contains specifications and performance verification information.
- *TSG4100A Series RF Signal Generator Installation and Safety Instructions*
  This manual provides basic installation information for getting started using the instrument. It also includes safety, compliance, and environmental information. It is available in both English and Russian.

Getting started

To help you get started with programming this instrument, this section includes the following subsections.

- *Overview of the manual*
  Summarizes each major section of this manual.
- *Connecting the interface*
  Describes interface options.
- *Using NI-VISA*
  Describes how to download the NI-VISA communication protocol.

Overview of the manual

The information contained in each major section of this manual is described below.

Syntax and Commands

Syntax and Commands, describes the structure and content of the messages your program sends to the generator. This section also describes the effect of each command and provides examples of how you might use it.

The *Command Groups* subsection provides lists by functional areas. The commands are listed alphabetically in the *Command Descriptions* section.

Status and Events

The program may request information from the instrument. The instrument provides information in the form of status and error messages. The *Status and Events* section describes how to get status or event information from the program and details the event and error messages.

Connecting the interface

You can use the LAN connector or the GPIB or RS-232 interfaces to send/receive remote commands.

When using a GPIB or RS-232 interface, remember the following:

- Assign a unique device address to each device on the bus. No two devices can share the same device address.
- Do not connect more than 15 devices to any one bus.
- Connect one device for every 2 m (6 ft) of cable used.
- Do not use more than 20 m (65 ft) of cable to connect devices to a bus.
- Turn on at least 2/3 of the devices on the network while using the network.
- Connect the devices on the network in a star or linear configuration, as shown in the following figure. Do not use loop or parallel configurations.

Using NI-VISA

NI (National Instruments) VISA (Virtual Instrument Software Architecture) provides a common standard for software developers so that software from multiple vendors, such as instrument drivers, can run on the same platform. You can use this software to write (or draw) interoperable instrument drivers in a variety of Application Development Environments (ADEs). It implements a subset of the VISA specification for controlling GPIB and serial (RS-232) instrument interfaces locally or remotely through an Ethernet LAN connection.

Installation

Use an internet browser to access the National Instruments Web site and download the current version of NI-VISA to your PC. Unzip the downloaded file in a temporary directory of your choice and run *Setup.exe*.

Syntax and Commands

Command Syntax

This section contains information on the Standard Commands for Programmable Instruments (SCPI) and IEEE 488.2 Common Commands you can use to program your TSG4100A Series generator. The information is organized in the following subsections:

- Backus-Naur Form Definition
- Syntax
- IEEE 488.2 Common Commands

Backus-Naur Form Definition

This manual may describe commands and queries using the Backus-Naur Form (BNF) notation. The following table defines the standard BNF symbols.

Table 2-1: BNF symbols and meanings

Symbol	Meaning
< >	Defined element
:=	Is defined as
|	Exclusive OR
{ }	Group; one element is required
[ ]	Optional; can be omitted
( )	Comment

NOTE. *Parameters (elements) listed without any surrounding characters are always required.*

Commands are terminated by a semicolon, a <CR> (ASCII 13), or a <LF> (ASCII 10). If the communications interface is GPIB, then the terminating character may optionally be accompanied by an EOI signal. If the EOI accompanies a character other than a <LF>, a <LF> will be appended to the command to terminate it. Execution of the command does not begin until a command terminator is received.

Syntax

All commands use ASCII characters, are four characters long, and are case-insensitive. Standard IEEE-488.2 defined commands begin with the '*' character followed by three letters. Instrument specific commands are composed of four letters.

The four letter mnemonic (shown in capital letters) in each command sequence specifies the command. The rest of the sequence consists of parameters. For example:

```
DELW 1
```

Sets and queries

Commands may take either set or query form, depending on whether the '?' character follows the mnemonic. Set only commands are listed without the '?', query only commands show the '?' after the mnemonic, and query optional commands are marked with a '(?)'.

Set commands specify actions for the instrument to perform. Query commands return measurement data and information about parameter settings.

NOTE. *Do not send () or {} or [] or spaces as part of the command.*

Query responses

When a query is sent to the instrument, only the values are returned, as shown in the following table.

Table 2-2: Query response examples

Query	Response
CATL?	0,512,1,0,2,128,3,0,4,0,5,0,6,0,7,0,8,0,9,0,10, 173696,11,720,12,128,13,79,14,235,15, 76800, 1044480
CNST?	2

A few queries also initiate an operation action before returning information. For example, the ERAS? query erases all user waveforms and then returns the free nonvolatile memory in 16-bit words.

Errors and failures

Aside from communication errors, commands may fail due to either syntax or execution errors. Syntax errors can be detected by looking at bit 5 (CME) of the event status register (*ESR?). Execution errors can be detected by looking at bit 4 (EXE) of the event status register. In both cases, an error code, indicating the specific cause of the error, is appended to the error queue. The error queue may be queried with the LERR? command. Descriptions of all error codes can be found in the section Error Codes. (See page 3-5.)

The command buffer is limited to 768 bytes, with 25 byte buffers allocated to each of up to 3 parameters per command. If the command buffer overflows, both the input and output buffers will be flushed and reset. If a parameter buffer overflows, a command error will be generated and the offending command discarded.

Parameter conventions

The command descriptions use parameters, such as i, f, and v. These parameters represent integers or floating point values expected by the command. The parameters follow the conventions summarized in the following table.

Table 2-3: Available units and parameter conventions

Parameter	Meaning
Arbitrary block [1]	A specified length of arbitrary data. #512234xxxxx . . . where 5 indicates that the following 5 digits (12234) specify the length of the data in bytes; xxxxx ... indicates the data
Hexadecimal [2]	Hexadecimal numbers (0-9, A, B, C, D, E, F). #HAA, #H1
i, j, k	An integer value.
d	A floating point value.
f	A floating point value representing a frequency in Hz.
p	A floating point value representing a phase in degrees.
t	A floating point value representing time in seconds.
v	A floating point value representing voltage in volts.
u	An identifier of units. Allowed units are:

Type	Allowed units
Amplitude	‘dBm’, ‘rms’, ‘Vpp’
Frequency	‘GHz’, ‘MHz’, ‘kHz’, or ‘Hz’
Time	‘ns’, ‘us’, ‘ms’, or ‘s’

1 Defined in ANSI/IEEE 488.2 as "Definite Length Arbitrary Block Response Data."

2 An ANSI/IEEE 488.2-1992-defined parameter type.

3 Some commands and queries will accept an octal or hexadecimal value even though the parameter type is defined as NR1.

You can omit a unit in a command, but you must include the unit when using a SI prefix. For example, frequency of 15 MHz can be described as follows

15.0E6, 1.5E7Hz, 15000000, 15000000Hz, 15MHz, etc.
("15M" is not allowed.)

Note that you can use either lower or upper case units and prefixes. The following examples have the same result, respectively.

170mhz, 170mHz, 170MHz, etc.
250mv, 250mV, 250MV, etc.

Numeric conventions

Floating point values may be decimal (‘123.45’) or scientific (‘1.2345e2’). Integer values may be decimal (‘12345’) or hexadecimal (‘0x3039’).

General Rules

Here are three general rules for using commands, queries, and parameters:

- You can use upper case, lower case, or a mixture of both cases for all commands, queries, and parameters.

  ```
  FDEV 10 kHz
  ```

  is the same as

  ```
  FDEv 10 kHz
  ```

IEEE 488.2 Common Commands

Description

ANSI/IEEE Standard 488.2 defines the codes, formats, protocols, and usage of common commands and queries used on the interface between the controller and the instruments. The generator complies with this standard.

Command and Query Structure

The syntax for an IEEE 488.2 common command is an asterisk (*) followed by a command and, optionally, a space and parameter value. The syntax for an IEEE 488.2 common query is an asterisk (*) followed by a query and a question mark. All of the common commands and queries are listed in the last part of the *Syntax and Commands* section. The following are examples of common commands:

- *ESE 16
- *CLS

The following are examples of common queries:

- *ESR?
- *IDN?

Command Groups

This section lists the instrument commands in two ways. It first presents them by functional groups. It then lists them alphabetically. The functional group list starts below. The alphabetical list provides more detail on each command.

Items followed by question marks are queries; items without question marks are commands. Some items in this section have a question mark in parentheses () in the command header section; this indicates that the item can be both a command and a query.

For the conventions of notation in this manual, refer to *Command Syntax* and following pages.

Functional Groups

All commands are divided into groups as shown in the following table.

Table 2-4: Command groups

Command group	Function
Common	Conforms to the IEEE Std 488.2.
Status	Controls the status and event registers.
Signal Synthesis	Controls signal parameters.
Modulation	Controls the modulation settings and parameters.
Interface	Controls interface settings and parameters.

The following section lists the commands by group.

Common Commands

The IEEE 488.2 common commands have a "*" prefix.

Table 2-5: Common commands

Header	Description
*CAL?	Performs an internal self-calibration.
*CLS	Clears status.
*ESE(?){i}	Sets or queries the bits in the SESE register.
*ESR?	Returns the contents of the SESR register.
*IDN?	Returns the instrument identification code.
*OPC(?)	Synchronizes commands.
*PSC(?){i}	Clears power on status.
*RCL i	Recalls instrument settings.
*RST	Returns the instrument settings to the factory defaults.
*SAV i	Saves instrument settings.
*SRE(?){i}	Sets or queries the bits in the SRER register.
*STB?	Returns the contents of the SBR using the MSS bit.
*TRG	Generates a trigger.
*TST?	Self test.
*WAI	Prevents the generator from executing further commands.

Interface Commands

Use the Interface commands to control instrument interface actions.

Table 2-6: Interface commands

Header	Description
EMAC?	Query the Ethernet MAC address.
EPHY(?){i}	Set (query) the Ethernet link speed.
IFCF(?)i{,j}	Set (query) interface configuration parameter.
IFRS i	Reset interface.
IPCF? i	Query active TCP/IP configuration parameter.
LOCK?	Request the instrument lock.
UNLK?	Release the instrument lock previously acquired by the LOCK? command.
XTRMi{,j,k}	Set the interface terminator that is appended to each response.

Modulation commands

Use the modulation commands to configure different types of modulation of the carrier. Basic configuration for analog modulation and vector modulation can be achieved by following the steps outlined in the related tables. (See Table 2-8.) (See Table 2-9.)

Table 2-7: Modulation commands

Header	Description
ADEP	Set (query) the AM modulation depth in percent.
ALPH	Set (query) the excess bandwidth factor, α, for Nyquist and Root-Nyquist filters.
ANDP	Set (query) the AM noise modulation depth {to d} in percent.
AWGN	Set (query) the current configuration for Additive White Gaussian Noise (AWGN).
BITS?	Query the current bits/symbol for the current constellation.
BTEE	Set (query) the current bandwidth, symbol time product, BT, for Gaussian filters.
CATL?	Query the current catalog listing of available user waveforms.
CNST	Set (query) the current user constellation.
COUP	Set (query) the coupling of the external modulation input.
DELW?	Delete the user waveform at the specified location and return an error code followed by the current free space in nonvolatile memory in 16-bit words.
ERAS?	Erase all user waveforms and return an error code followed by the current free space in nonvolatile memory in 16-bit words.
FDEV	Set (query) the FM deviation.
FLTR	Set (query) the current pulse shaping filter.
FNDV	Set (query) the FM noise deviation.
MFNC	Set (query) the modulation function or AM/FM/ΦM.
MODI	Set (query) the modulation index for CPM.
MODL	Set (query) the enable state of modulation.
MPRE	Configure the instrument according to the given modulation preset.
NPWR	Set (query) the current AWGN power in dB.
OFSI	Set (query) the current offset for I in IQ modulation in percent.
OFSQ	Set (query) the current offset for Q in IQ modulation in percent.
PDEV	Set (query) the ΦM deviation in degrees.
PDTY	Set (query) the duty factor for pulse modulation in percent.
PFNC	Set (query) the modulation function for pulse modulation.
PNDV	Set (query) the ΦM noise deviation in degrees.

Table 2-7: Modulation commands (cont.)

Header	Description
PPER	Set (query) the pulse modulation period.
PRBS	Set (query) the PRBS length for pulse modulation.
PTRN	Set (query) the 16-bit data word for pattern waveforms.
PWID	Set (query) the pulse modulation width (duty cycle).
QFNC	Set (query) the modulation function for IQ modulation.
RATE	Set (query) the modulation rate for AM/FM/PM.
RPER	Set (query) the PRBS period for pulse modulation.
SAVC?	Save the user constellation stored in SRAM to the specified location in nonvolatile memory and return an error code when complete.
SAVF?	Save the user filter stored in SRAM to the specified location in nonvolatile memory and return an error code when complete.
SAVW?	Save the user waveform stored in SRAM to the specified location in nonvolatile memory and return an error code when complete.
SCAL	Set (query) the digital scale factor for modulation waveforms.
STAG?	Query whether or not the current constellation operates in staggered mode.
STYP	Set (query) the modulation subtype.
SYMR	Set (query) the symbol rate for digital waveforms.
TDMA?	Query the current TDMA configuration word.
TYPE	Set (query) the current modulation type.
WAVF	Set (query) the current user waveform.
WRTC	Write the given user constellation to SRAM.
WRTE	Write the given event marker configuration to SRAM.
WRTF	Write the given filter to SRAM.
WRTW	Write the given user waveform to SRAM.

Basic analog modulation configuration

Basic configuration for analog modulation can be achieved by following the steps outlined in the following table. The remote interface commands associated with each setting are shown. Analog modulation is indicated by setting the subtype to zero (STYP command). Modulation type is selected with the TYPE command.

Although not shown in this table, analog modulation also supports user waveforms downloaded by the user. This is achieved by setting the modulation function (MFNC or PFNC) to user waveform and selecting the desired user waveform with the WAVF command.

Table 2-8: Basic analog modulation configuration

Modulation	Configuration	Relevant commands
On/Off	Enable configuration	MODL
External	AC/DC input coupling	COUP
AM	Select AM modulation	TYPE 0, STYP 0
	Modulation function	MFNC
	Modulation rate / Noise bandwidth	RATE
	Deviation	ADEP, ANDP
FM	Select FM modulation	TYPE 1, STYP 0
	Modulation function	MFNC
	Modulation rate / Noise bandwidth	RATE
	Deviation	FDEV, FNDV
ΦM	Select ΦM modulation	TYPE 2, STYP 0
	Modulation function	MFNC
	Modulation rate / Noise bandwidth	RATE
	Deviation	PDEV, PNDV
Pulse	Select pulse	TYPE 4
	Modulation function	PFNC
	Pulse period	PPER
	Pulse width	PWID or PDTY
	PRBS period	RPER
	PRBS length	PRBS

Basic vector modulation configuration

Basic configuration for digital vector modulation can be achieved by following the steps outlined in the following table.

The remote interface commands associated with each setting are shown in this table. In contrast to analog modulation, digital vector modulation involves the mapping of digital symbols onto a vector constellation. Thus, instead of a waveform frequency or period, there is a symbol rate (SYMR command). The digital symbols are also typically played back through pulse shaping filter characterized through the FLTR command.

Table 2-9: Digital vector modulation configuration

Modulation	Configuration	Relevant commands
On/Off	Enable configuration	MODL
ASK	Select AM modulation	TYPE 0
	Select constellation	STYP, CNST
	Modulation function	QFNC, WAVF
	Pulse shaping filter	FLTR, ALPH, BTEE
	Symbol rate	SYMR
FSK	Select FM modulation	TYPE 1
	Select constellation	STYP, CNST
	Modulation function	QFNC, WAVF
	Pulse shaping filter	FLTR, ALPH, BTEE
	Symbol rate	SYMR
	Deviation	FDEV
PSK	Select PSK modulation	TYPE 2
	Select constellation	STYP, CNST
	Modulation function	QFNC, WAVF
	Pulse shaping filter	FLTR, ALPH, BTEE
	Symbol rate	SYMR
QAM	Select QAM modulation	TYPE 7
	Select constellation	STYP, CNST
	Modulation function	QFNC, WAVF
	Pulse shaping filter	FLTR, ALPH, BTEE
	Symbol rate	SYMR
CPM	Select CPM modulation	TYPE 8
	Select constellation	STYP
	Modulation function	QFNC, WAVF
	Pulse shaping filter	FLTR, ALPH, BTEE
	Symbol rate	SYMR
	Modulation index	MODI
VSB	Select VSB modulation	TYPE 9
	Select constellation	STYP
	Modulation function	QFNC, WAVF
	Pulse shaping filter	FLTR, ALPH, BTEE
	Symbol rate	SYMR

Signal Synthesis Commands

Use the signal synthesis commands to set the frequency, amplitude, and phase of the outputs. Basic signal configuration can be achieved by following the steps as outlined in the related table. (See Table 2-11.)

Table 2-10: Signal synthesis commands

Header	Description
AMPL	Set (query) the amplitude of the low frequency BNC output.
AMPR	Set (query) the amplitude of the Type N RF output.
ENBL	Set (query) the enable state of the low frequency BNC output.
ENBR	Set (query) the enable state of the Type N RF output.
FREQ	Set (query) the carrier frequency.
NOIS	Set (query) the RF PLL loop filter mode for the instrument.
OFSL	Set (query) the offset voltage of the low frequency BNC output in V.
PHAS	Set (query) the phase of the carrier.
RPHS	Set the current phase of the carrier to 0°.

Basic signal configuration

Basic signal configuration can be achieved by following the steps as outlined in the following table.

Table 2-11: Basic signal configuration

Action	Relevant commands
Set frequency	FREQ
Set amplitude	AMPL, AMPR
Set offset	OFSL
Adjust phase	PHAS, RPHS

Status Commands

Use these commands to control registers defined in the SCPI status reporting structure. You can read more about status registers in the related section.

Table 2-12: Status commands

Header	Description
INSE	Set (query) the Instrument Status Enable register.
INSR?	Query the Instrument Status Register. Upon executing a INSR? query, the returned bits of the INSR register are cleared.
LERR?	Query the last error in the error buffer. Upon executing a LERR? query, the returned error is removed from the error buffer.
ORNG?	Query whether specified output is over its specified range.
TEMP?	Query the current temperature of the RF output block in degrees C.
TIMB?	Query the current timebase.

Retrieving Response Message

When receiving a query command from the external controller, the generator puts the response message on the Output Queue. This message cannot be retrieved unless you perform retrieval operations through the external controller. (For example, call the IBRD subroutine included in the GPIB software of National Instruments.)

When the Output Queue contains a response message, sending another command from the external controller before retrieving this message deletes it from the queue. The Output Queue always contains the response message to the most recent query command.

You can use the MAV bit of the Status Byte Register (SBR) to check whether the Output Queue contains a response message. For details, refer to *Status Byte Register (SBR)*.

Command Descriptions

ADEP

Sets or queries the AM modulation depth {d} in percent.

Group Modulation commands

Syntax
```
ADEP {d}
```

Related Commands See ANDP if noise is the selected modulation function.

Arguments {d} is a floating point value.

Returns A floating point value.

Examples `ADEP 90.0` sets the depth to 90%.

`ADEP?` might return 85.0, indicating that the depth is 85%.

ALPH

Sets or queries the excess bandwidth factor, α, for the nyquist and root-nyquist filters.

Conditions Range: 0.1 to 1.0

Group Modulation commands

Syntax
```
ALPH {d}
ALPH?
```

Arguments {d} is a floating point value.

Returns A floating point value.

Examples `ALPH 0.2` sets α to 0.2.

`ALPH?` might return 0.5, indicating that the α value is 0.5.

AMPL

Sets or queries the amplitude of the low frequency BNC output {v}.

Conditions If units are omitted, they default to dBm.

Group Signal synthesis commands

Syntax
```
AMPL {v}
AMPL?
```

Arguments {v units} where v is a floating point value and units can be RMS or nothing (defaulting to dBm).

Returns A floating point value.

Examples `AMPL −1.0` sets the BNC output amplitude to –1.0 dBm.

`AMPL 0.1 RMS` sets the BNC output amplitude to 0.1 V_{rms}.

`AMPL?` might return 0.2, indicating that the amplitude is 0.2 V_{rms}.

AMPR

Sets or queries the amplitude of the Type N RF output.

Conditions If omitted, units default to dBm.

Group Signal synthesis commands

Syntax
```
AMPR {v}[u]
AMPR?
```

Arguments {v} = amplitude value as a floating point value without an exponent.

[u] = unit of measure (RMS, dBm)

Returns A floating point value.

Examples `AMPR -3.0` sets the output amplitude to –3.0 dBm.

`AMPR 0.1 RMS` sets the output amplitude to 0.1 V_{rms}.

`AMPR?` might return 0.1, indicating that the output amplitude is 0.1 V_{rms}.

ANDP

Sets or queries the AM noise modulation depth {d} in percent. The value controls the rms depth of the modulation (not the peak deviation, as the `ADEP` command does).

Group Modulation commands

Syntax
```
ANDP {d}
ANDP?
```

Related Commands See the ADEP command for all modulation functions other than noise.

Arguments {d} is a floating point value.

Returns A floating point value.

Examples `ANDP 10.0` sets the rms noise depth to 10%.

`ANDP?` might return 9.0, indicating that the rms noise depth is 9%.

AWGN

Sets or queries the current configuration for AWGN {i}.

Group Modulation commands

Syntax
```
AWGN {i}
AWGN?
```

Related Commands See NPWR to configure the noise power.

Arguments

0 = Noise off

1 = Noise added

2 = Noise only

Returns

0 = noise is off

1 = noise is added

2 = noise only

Examples

`AWGN 1` sets the instrument to add noise.

`AWGN?` might return `0`, indicating that the noise is off.

BITS? (Query Only)

Queries the current bits/symbol for the current constellation. Bits/symbol is set by the selected constellation.

Conditions

IQ waveforms that contain 16-bit IQ value pairs will return a value of 16 or 32.

If constellations are not used, the return value will be 16 or 32.

Group

Modulation commands

Syntax

`BITS?`

Returns

An integer.

Examples

`BITS?` might return 16, indicating that the current bits/symbol is 16.

BTEE

Sets or queries the current bandwidth symbol time product (BT) for Gaussian filters.

Conditions

Range: 0.1 to 1.0

Group

Modulation commands

Syntax BTEE

Arguments {d} is a floating point value.

Returns A floating point value.

Examples BTEE 0.3 sets BT to 0.3.

BTEE? might return 0.3, indicating that the value for BT is 0.3.

*CAL? (Query Only)

This command currently does nothing and returns 0.

Group Common commands

Syntax *CAL?

Returns 0

Examples *CAL? does nothing and will return 0.

CATL? (Query Only)

Queries a catalog listing of User waveforms. Returns a comma separated list of location and waveform size in 16-bit words. When all waveforms have been itemized, the current free space in 16-bit words for new waveforms is appended to the list.

Group Modulation commands

Syntax CATL?

Returns Delta marker amplitude for the selected marker. See Examples.

Examples CATL? might return 0,512,1,0,2,128,3,0,4,0,5,0,6,0,7,0,8,0,9,0,10,173696,11,720,12,128,13,79,14,235,15,76800,1044480. In this example, SRAM has a

waveform 512 words long. Location 2 has a waveform 128 words long. All other user waveforms are empty. Locations 10 to 14 indicate the presence of factory loaded read-only waveforms (see WAVF for details). Finally, the last number indicates 1,044,480 words of free space still available for storage of new waveforms in nonvolatile memory.

*CLS (No Query Form)

Immediately clears the ESR and INSR registers, as well as the LERR error buffer.

Group Common commands

Syntax

```
*CLS
```

Arguments None

Examples `*CLS` will immediately clear the ESR and INSR registers and the LERR error buffer.

CNST

Sets or queries the current user constellation. Note that user constellations are only active when the modulation subtype is user (see STYP command).

Conditions Range for {i}: 0 to 9.

The value, 0, identifies the constellation stored in SRAM. Values 1 to 9 identify user constellations stored in nonvolatile memory using the SAVC command.

Group Modulation commands

Syntax

```
CNST {i}
CNST?
```

Related Commands STYP

SAVC?

Arguments {i} is a constellation identified as 0 through 9.

Examples

```
CNST

CNST?
```

COUP

Sets or queries the coupling of the external modulation input. If i is 0, the input is AC coupled. If i is 1, the input is DC coupled. This setting has no affect on the input if pulse modulation is active. In that case, the coupling is always DC.

Group Modulation commands

Syntax

```
COUP {i}
COUP?
```

Arguments

0 sets external modulation input to AC coupled.

1 sets external modulation input to DC coupled.

Returns

0 means external modulation input is set to AC coupled.

1 means external modulation input is set to DC coupled.

DELW? (Query Only)

Deletes the user waveform at location i and returns an error code followed by the current free space in nonvolatile memory in 16-bit words. Factory loaded, read-only waveforms cannot be deleted with this command.

CAUTION. *Once deleted, a user waveform cannot be recovered.*

Conditions Range for i: 0 to 9.

Group Modulation commands

Syntax

```
DELW?{i}
```

Related Commands SAVW?

Arguments 0 identifies any waveform stored in SRAM.

1 to 9 identify waveforms stored in nonvolatile memory through the SAVW command.

Returns Integer value in 16-bit words.

An error code 0 means the delete was successful.

Examples `DELW?` might return 0, 1040384\r\n, indicating that the delete was succefull (0) and the current free space in nonvolatile memory.

EMAC? (Query Only)

Queries the Ethernet MAC address.

Group Interface commands

Syntax

```
EMAC?
```

Returns Floating point value with decimal.

Examples `EMAC?` might return 134.023.001.001.

ENBL

Sets or queries the enable state of the low frequency BNC output.

Group Signal synthesis commands

Syntax

```
ENBL {i}
ENBL?
```

Arguments 0 = the output is disabled and turned off.

1 = the output is enabled and operating at the programmed amplitude for the output.

Returns 0 when the output is disabled.

1 when the output is enabled.

NOTE. *0 may be returned even if a 1 was sent if the output is not active at the current frequency (i.e., $F_{carrier}$ > 62.5 MHz).*

Examples `ENBL 1` will enable the output.

`ENBL?` might return 0, indicating that the output is disabled.

ENBR

Sets or queries the enable state of the low frequency Type N RF output.

Group Signal synthesis commands

Syntax

```
ENBR {i}
ENBR?
```

Arguments 0 = the output is disabled and turned off.

1 = the output is enabled and operating at the programmed amplitude for the output.

Returns 0 when the output is disabled.

1 when the output is enabled.

NOTE. *0 may be returned even if a 1 was sent if the output is not active at the current frequency (i.e., $F_{carrier}$ <950 kHz).*

Examples `ENBR1` will enable the output.

`ENBR?` might return 0, indicating that the output is disabled.

EPHY

Sets or queries the Ethernet link speed.

Group Interface commands

Syntax

```
EPHY {i}
EPHY?
```

Arguments

0 = 10 Base T

1 = 100 Base T

Returns

0 means link speed is 10 Base T

1 means link speed is 100 Base T

Examples

`EPHY 1` will configure the link speed for 100 Base T operation.

ERAS? (Query Only)

Erases all user waveforms and returns an error code of 0, if successful, followed by the current free space in nonvolatile memory in 16-bit words. Factory loaded, read-only waveforms cannot be deleted with this command.

CAUTION. *Once deleted, user waveforms cannot be recovered.*

Group

Modulation commands

Syntax

```
ERAS?
```

Related Commands

DELW?

Returns

0,16-bit words

Examples

`ERAS?` might return `0,1048576`, indicating that the erase was successful (0) and then the size if the current free space in nonvolatile memory.

*ESE

Sets or queries the Standard Event Status Enable register. Bits set in this register cause ESB (in STB) to be set when the corresponding bit is set in the ESR register.

Group

Common commands

Syntax

```
*ESE {i}
*ESE?
```

Related Commands *ESR?

Arguments 0 - 7

Refer to the *Standard Event Status Register* section for the meaning of these bits in the ESER register. (See page 3-2.)

Examples `*ESE 176` sets the PON, CME, and EXE bits.

*ESR? (Query Only)

Queries the Standard Event Status Register. Upon executing a *ESR? query, the returned bits of the *ESR register are cleared.

Group Common commands

Syntax

```
*ESR?
```

Related Commands *ESE

Returns 0 - 7

Refer to the *Standard Event Status Register* section for the meaning of these bits in the ESR register. (See page 3-2.)

Examples `*ESR?` might return 176, indicating that PON, CME, and EXE are set.

FDEV

Sets or queries the FM deviation. If omitted, units default to Hz.

Group Modulation commands

Syntax

```
FDEV {f}
FDEV?
```

Related Commands FNDV: Use this command if noise is the selected modulation function.

Arguments A floating point value that is the fm deviation.

Returns An integer that is the FM deviation.

Examples `FDEV 10e3` sets the FM deviation to 10 kHz.

`FDEV?` might return 1000, indicating that the current FM deviation is 1000 Hz.

`FDEV1 kHz` sets the FM deviation to 1 kHz.

FLTR

Sets or queries the current pulse shaping filter.

Group Modulation commands

Syntax

```
FLTR {i}
FLTR?
```

Related Commands FDEV

ALPH

BTEE

Arguments {i}= the pulse shaping filter. It may be set to one of the following values:

i	Filter
0	User filter stored in SRAM.
1-9	User filters stored in nonvolatile memory.
10	Raised cosine filter.
11	Root-raised cosine filter.
12	Gaussian filter.
13	Rectangular filter.
14	Triangular filter (simple, linear interpolation).
15	Kaiser windowed sinc filter.
16	Linearized Gaussian filter (used in GSM EDGE modulation).
17	C4FM (used in APCO 25 modulation).

For raised cosine and root-raised cosine filters, see the ALPH command to set the excess bandwidth factor, α. For the Gaussian filter, see the BTEE command to set the bandwidth, symbol time product, BT.

Returns See table under Arguments.

Examples `FLTR 10` sets the pulse shaping filter to raised cosine.

`FLTR?` might return 17, indicating that the pulse shaping filter is set to C4FM.

`ALPH 0.3` sets αfor raised cosine filter to 0.3.

FNDV

Sets or queries the FM noise deviation. If omitted, units default to Hz. The value controls the rms deviation of the modulation, not the peak deviation as the FDEV command does.

Group Modulation commands

Syntax
```
FNDV {f}
FNDV?
```

Related Commands See the FDEV command for all modulation functions other then noise.

Arguments A floating point value that is the rms deviation.

Returns An integer that is the FM noise deviation.

Examples `FNDV 10e3` sets the rms noise deviation to 10 kHz.

`FNDV?` might return 1000, indicating the rms FM noise deviation is 1 kHz.

`FNDV 1 kHz` sets the rms FM noise deviation to 1 kHz.

FREQ

Sets or queries the carrier frequency.

Conditions If omitted, units default to Hz.

Group Signal synthesis commands

Syntax
```
FREQ {f}[u]
FREQ?
```

Arguments {i} is a floating point value.

Returns An floating point value that is the carrier frequency.

Examples `FREQ 100000000.000000` sets the carrier frequency to 100 MHz.

`FREQ 100 MHz` sets the carrier frequency to 100 MHz.

`FREQ ?` might return 100e6, indicating that the carrier frequency is 100 MHz.

`FREQ? MHz` might return 100, indicating the carrier frequency is 100 MHz.

*IDN? (Query Only)

Queries the instrument identification string.

Group Common commands

Syntax
```
*IDN?
```

Returns String indicating the instrument manufacturer, model, serial number, and firmware version.

Examples `*IDN?` might return `Tektronix,TSG4106A,s/nQ000001,ver2.03.26`.

IFCF

Sets or queries the interface configuration i {to j}.

Group Interface commands

Syntax
```
IFCF i{,j}
IFCF?
```

Arguments Set j to 0 to disable a setting.

Set j to 1 to enable a setting.

Set i as follows:

i	Parameter
0	RS-232 Enable/Disable
1	RS-232 Baud Rate
2	GPIB Enable/Disable
3	GPIB Address
4	LAN TCP/IP Enable/Disable
5	DHCP Enable/Disable
6	Auto-IP Enable/Disable
7	Static IP Enable/Disable
8	Bare Socket Enable/Disable
9	Telnet Enable/Disable
10	VXI-11 Net Instrument Enable/Disable
11	Static IP Address
12	Subnet Address/Network Mask
13	Default Gateway

Valid RS-232 baud rates include 4800, 9600, 19200, 38400, 57600, and 115200. Valid GPIB addresses are in the range 0-30. Parameters 10-12 require an IP address in the form ‘a.b.c.d’ where each letter is a decimal integer in the range 0-255.

Returns See Arguments.

Examples `IFCF 6.0` disables the Auto IP setting.

`IFCF 1,19200` sets RS-232 baud rate to 19200.

`IFCF 3,16` sets the primary GPIB address to 16.

`IFCF 11,192.168.10.5` sets IP Address to 192.168.10.5.

IFRS (No Query Form)

Resets the specified interface.

Group Interface commands

Syntax `IFRS i`

Arguments i is:

0 = RS-232

1 = GPIB

2 = LAN TCP/IP

INSE

Sets or queries the Instrument Status Enable register. Bits set in this register cause INSB (in STB) to be set when the corresponding bit is set in the INSR register.

Group Status commands

Syntax
```
INSE {i}
INSE?
```

Related Commands INSR?

Arguments {i} is the bits to which to set the instrument SER. The bits in the INSR register have the following meaning:

Bit	Name	Description
0	20MHZ_UNLK	20 MHz PLL unlocked.
1	100MHZ_UNLK	100 MHz PLL unlocked.
2	19MHZ_UNLK	19 MHz PLL unlocked.
3	1GHZ_UNLK	1 GHz PLL unlocked.
4	4GHZ_UNLK	4 GHz PLL unlocked.
5	NO_TIMEBASE	Installed timebase is not oscillating.
6	RB_UNLOCK	The installed Rubidium oscillator is unlocked.
7	—	Reserved
8	MOD_OVLD	Modulation overloaded.
9	IQ_OVLD	IQ modulation overloaded.
10-15	—	Reserved

Returns See Arguments.

INSR? (Query Only)

Queries the Instrument Status Register (INSR). Upon executing a INSR? query, the returned bits of the INSR register are cleared.

Group Status commands

Syntax `INSR?`

Related Commands INSE

Returns The bits in the INSR register have the following meaning:

Bit	Name	Description
0	20MHZ_UNLK	20 MHz PLL unlocked.
1	100MHZ_UNLK	100 MHz PLL unlocked.
2	19MHZ_UNLK	19 MHz PLL unlocked.
3	1GHZ_UNLK	1 GHz PLL unlocked.
4	4GHZ_UNLK	4 GHz PLL unlocked.
5	NO_TIMEBASE	Installed timebase is not oscillating.
6	RB_UNLOCK	The installed Rubidium oscillator is unlocked.
7	—	Reserved
8	MOD_OVLD	Modulation overloaded.
9	IQ_OVLD	IQ modulation overloaded.
10-15	—	Reserved

Examples `INSR?` might return 08, indicating that a modulation overload was detected and the 20 MHz PLL came unlocked.

IPCF? (Query Only)

Queries the active TCP/IP configuration.

Group Interface commands

Syntax `IPCF? i`

Arguments i is:

0 = Link

1 = IP Address

2 = Subnet Address/Network Mask

3 = Default Gateway

The link parameter indicates whether the unit is physically connected to the LAN/Ethernet network. A value of 1 indicates the unit is connected. The rest of the parameters indicate the current TCP/IP configuration that was selected by the appropriate configuration process: DHCP, Auto-IP, or Static IP.

LERR? (Query Only)

Queries the last error in the error buffer. Upon executing a LERR? query, the returned error is removed from the error buffer.

NOTE. *Read about Error Codes for a description of the possible error codes returned by this command. The error buffer has space to store up to 20 errors. If more than 19 errors occur without being queried, the 20th error will result in error 254 (Too Many Errors), indicating that errors were dropped.* *(See page 3-5.)*

Group Status commands

Syntax LERR?

Returns (See page 3-5, *Error messages and codes*.)

LOCK? (Query Only)

Requests the instrument lock and returns the result. When the lock is granted, no other instrument interface, including the front panel interface, may alter instrument settings until the lock is released with the UNLK command.

Group Interface commands

Syntax LOCK?

Related Commands UNLK?

Returns

0 means the lock was unsuccessful.

1 means the lock was successful.

MFNC

Sets or queries the modulation function or AM/FM/ΦM.

Group

Modulation commands

Syntax

```
MFNC {i}
MFNC?
```

Related Commands

See the following commands for pulse (PFNC) or IQ (QFNC) modulations.

PFNC

QFNC

Arguments

The parameter i may be set to one of the following values:

i	Modulation function
0	Sine wave
1	Ramp
2	Triangle
3	Square
4	Noise
5	External
11	User waveform

Returns

See Arguments.

Examples

`MFNC 0` sets the modulation function to sine wave.

`MFNC?` might return 11, indicating that the modulation function is set to user waveform.

MODI

Sets or queries the modulation index for CPM. The modulation index is stored as a 3 digit floating point decimal. However, when applied to the modulation, it will be rounded to the nearest rational factor, n/512, where n is an integer.

Conditions Range: 0.0 to 1.0

Group Modulation commands

Syntax
```
MODI {d}
MODI?
```

Arguments {d} is a 3 digit floating point decimal.

Returns A floating point value.

Examples `MODI 0.5` sets the modulation index to 1/2.

`MODI 0.438` sets the modulation index to 7/16.

`MODI?` might return 0.5, indicating that the modulation index is 1/2.

MODL

Sets or queries the enable state of modulation. This command may fail if the current modulation type is not allowed at current settings. For example, pulse modulation is not allowed at frequencies where the RF doubler is active.

Group Modulation commands

Syntax
```
MODL {i}
MODL?
```

Arguments 0 disables modulation.

1 enables modulation.

Returns 0 means modulation is disabled.

1 means modulation is enabled.

Examples `MODL 1` enables modulation.

`MODL?` might return 0, indicating that modulation is disabled.

MPRE (No Query Form)

Configures the instrument according to the given modulation preset.

Group Modulation commands

Syntax

```
MPRE i
```

Related Commands MODL

Arguments {i} identifies the modulation preset, which may be one of the following values:

i	Modulation preset
0	AM audio clip
1	FM audio clip
2	NADC modulation
3	PDC modulation
4	DECT frame
5	APCO 25 modulation
6	TETRA modulation
7	GSM frame
8	GSM EDGE frame
9	W-CDMA frame

Examples This command is used in a sequence, as shown in the following example:

`MPRE 8` sets the instrument to modulate a frame of GSM EDGE.

MODL `1` enables the GSM EDGE modulation set in the previous command.

NOIS

Sets or queries the RF PLL loop filter mode for the instrument.

Group Signal synthesis commands

Syntax

```
NOIS {i}
NOIS?
```

Arguments `0` is Mode 1, which minimizes noise at small offsets from carrier.

`1` is Mode 2, which minimizes noise at large offsets from carrier.

Returns `0` means the RF PLL loop is set to Mode 1.

`1` means the RF PLL loop is set to Mode 2.

NPWR

Sets or queries the current AWGN power in dB.

Conditions Noise power range: -10.0 to -70 dB

Group Modulation commands

Syntax
```
NPWR {d}
NPWR?
```

Related Commands See AWGN for configuring whether noise is to be added to a modulation.

Arguments {d} is a floating point value.

Returns A floating point value.

Examples `NPWR?` might return –50.0, indicating that the noise power of the current modulation is –50.0 dB.

This command can be used in a sequence, as shown in the following example:

`NPWR -30.0` sets the noise power to –30.0 dB.

AWGN `1` adds noise to the current modulation.

OFSI

Sets or queries the current offset for I in IQ modulation in percent. IQ offsets only apply to internally generated IQ waveforms

Conditions Offset range: –5.0% to +5.0%

Group Modulation commands

Syntax
```
OFSI {d}
OFSI?
```

Related Commands OFSQ

Arguments {d} is a floating point value.

Returns A floating point value that is current offset for I.

Examples `OFSI 0.5` sets the offset for I to 0.5%.

`OFSI -1.5` sets the offset for I to –1.5%.

OFSL

Sets or queries the offset voltage of the low frequency BNC output in V.

Group Signal synthesis commands

Syntax
```
OFSL {v}
OFSL?
```

Arguments {v} is a floating point value.

Returns A floating point value that is the offset voltage in Volts.

Examples `OFSL 1` sets the offset voltage to 1 V.

`OFSL?` might return –0.5, indicating the offset voltage.

OFSQ

Sets or queries the current offset for Q in IQ modulation in percent. IQ offsets only apply to internally generated IQ waveforms.

Conditions Offset range: –5.0% to +5.0%

Group Modulation commands

Syntax
```
OFSQ {d}
OFSQ?
```

Related Commands OFSI

Arguments {d} is a floating point value.

Returns A floating point value that is the current offset for Q.

Examples `OFSQ 0.5` sets the offset for Q to 0.5%.

`OFSQ-1.5` sets the offset for Q to –1.5%.

*OPC

Sets the OPC flag in the ESR register when all prior commands have completed.

The query form returns 1 when all prior commands have completed, but does not affect the ESR register.

Group Common commands

Syntax
```
*OPC
*OPC?
```

Arguments None.

Returns 1 means all prior commands have completed.

ORNG? (Query Only)

Queries whether output i is over its specified range.

Group Status commands

Syntax
```
ORNG? {i}
```

Arguments The parameter i identifies the output as follows. If omitted, i defaults to 0.

0 is the BNC output

1 is the Type N output

Returns 1 means that the given output is over range.

0 means the given out put is within range.

PDEV

Sets or queries the ΦM deviation in degrees.

Group Modulation commands

Syntax
```
PDEV {p}
PDEV?
```

Related Commands See PNDV command if noise is the selected modulation function.

Arguments {p} is a floating point value.

Returns A floating point value.

Examples `PDEV 45.0` sets the ΦM deviation to 45.0 degrees.

`PDEV?` might return 45.0, indicating that the current ΦM deviation is 45.0 degrees.

PDTY

Sets queries the duty factor for pulse modulation in percent. This value controls pulse modulation when the selected waveform is square (see `PFNC`). Use `PWID?` to determine the actual pulse width in time.

Group Modulation commands

Syntax
```
PDTY {d}
PDTY?
```

Related Commands PFNC

PWID

Arguments {d} is a floating point value / an integer.

Returns A floating point value / an integer.

Examples PDTY 10 sets the duty factor to 10%.

PDTY? might return 5, indicating that the current duty factor is 5%.

PFNC

Sets or queries the modulation function for pulse modulation.

Group Modulation commands

Syntax

```
PFNC {i}
PFNC?
```

Related Commands See MFNC for AM/FM/ΦM modulation.

See QFNC for IQ modulation.

Arguments The parameter {i} may be set to one of the following values:

i	Modulation function
3	Square
4	Noise (PRBS)
5	External
11	User waveform

Returns See Arguments.

Examples PFNC 3 sets the modulation function to Square.

PFNC? might return 5, indicating that the modulation function is External.

PHAS

Sets or queries the phase of the carrier.

The phase will track to ±360°, but it may only be stepped by 360° in one step. Thus, if the phase is currently 360°, setting the phase to –90° will fail because the phase step is larger than 360°. On the other hand, setting the phase to 370° will succeed, but the reported phase will then be 10°.

Group Signal synthesis commands

Syntax

```
PHAS {p}
PHAS?
```

Arguments {p} is a floating point value.

Returns A floating point value.

Examples `PHAS 90.0` sets the phase to 90 degrees.

`PHAS?` might return 10.0, indicating the phase is 10 degrees.

PNDV

Sets or queries the ΦM noise deviation in degrees. The value controls the rms deviation of the modulation, not the peak deviation as the `PDEV` command does.

Group Modulation commands

Syntax

```
PNDV {p}
PNDV?
```

Related Commands See PDEV for all modulation functions other than noise.

Arguments {p} is a floating point value.

Returns A floating point value.

Examples `PNDV 10.0` sets the ΦM deviation to 10.0 degrees.

`PNDV?` might return 10.0 degrees, indicating that the ΦM deviation is 10.0 degrees.

PPER

Sets or queries the pulse modulation period. If omitted, units default to seconds. This value controls pulse modulation when the selected waveform is square (see PFNC).

Group Modulation commands

Syntax
```
PPER {t}
PPER?
```

Arguments {t} is a floating point value.

Returns A floating point value.

Examples `PPER 1e-3` sets the pulse period to 1 ms.

`PPER?` might return 0.05, indicating the pulse period is 0.05 seconds.

PRBS

Sets or queries the PRBS length for pulse modulation. This value controls pulse modulation when the selected waveform is noise (see `PFNC`).

Conditions Range: 5 to 32

Group Modulation commands

Syntax
```
PRBS {i}
PRBS?
```

Related Commands PFNC

Arguments {i} is the number of bits in the PRBS generator.

A value of 8, for example, means the generator is 8 bits wide. It will generate a sequence of pseudo random bits that repeats every $2^8 - 1$ bits.

Returns See Arguments.

Examples `PRBS` 10 sets the PRBS length to 10.

`PRBS`? might return 5, indicating that the PRBS length is 5.

*PSC

Sets or queries the Power-on Status Clear flag. The Power-on Status Clear flag is stored in nonvolatile memory in the unit, and thus, maintains its value through power-cycle events.

Group Common commands

Syntax

```
*PSC {i}
*PSC?
```

Related Commands *ESE

*SRE

Arguments If the value of the flag is 0, then the Service Request Enable and Standard Event Status Enable Registers (*SRE, *ESE) are stored in non-volatile memory, and retain their values through power-cycle events.

If the value of the flag is 1, then these two registers are cleared upon power-cycle.

Returns 0 means the SRE and ESE are stored.

1 means the SRE and ESE are cleared at power-cycle.

Examples `*PSC 1` sets the power-on status clear to 1.

`*PSC?` might return 1, indicating that the power-on status clear is set to 1.

PTRN

Sets or queries the 16-bit data word for pattern waveforms.

Group Modulation commands

Syntax
```
PTRN {i}
PTRN?
```

Arguments {i} in hexadecimal.

Returns Hexadecimal value.

Examples `PTRN 0x1E1E` sets the current 16-bit data word to 0x1E1E in hex.

`PTRN?` might return 0x1E1E, indicating that is the current 16-bit data word.

PWID

Sets or queries the pulse modulation width (duty cycle). This value controls pulse modulation when the selected waveform is square (see `PFNC`).

Group Modulation commands

Syntax
```
PWID {t}[u]
```

Related Commands PFNC

Arguments {t} is a floating point value.

[u] (If omitted, units default to seconds.)

Returns A floating point value that is the pulse width in seconds.

Examples `PWID1e-6` sets the pulse width to 1 µs.

QFNC

Sets or queries the modulation function for IQ modulation.

Group Modulation commands

Syntax

```
QFNC {i}
QFNC?
```

Related Commands See MFNC for AM/FM/ΦM modulation.

See PFNC for pulse modulation.

Arguments The parameter i may be set to one of the following values:

{i}	Modulation function
0	Sine wave
1	Ramp
2	Triangle
3	Square
4	Phase noise
5	External
6	Sine/Cosine
7	Cosine/Sine
8	IQ noise
9	PRBS symbols
10	Pattern (16 bits)
11	User waveform

Not all values are valid in all modulation modes.

Returns See Arguments.

RATE

Sets or queries the modulation rate. If units are omitted, units default to Hz. This command also controls the noise bandwidth if a noise function is selected for the given type of modulation.

Group Modulation commands

Syntax

```
RATE {f}[u]
RATE?
```

Related Commands

See SYMR to control the symbol rate of user waveforms.

Arguments

{f} is a floating point value.

Returns

A floating point value.

Examples

`RATE400` sets the modulation rate to 400 Hz.

`RATE10 kHz` sets the rate to 10 kHz.

`RATE?` might return 400, indicating that the modulation rate is 400 Hz.

`RATE? kHz` might return 10, indicating the rate is 10 kHz.

*RCL (No Query Form)

Recalls instrument settings from the specified location.

Conditions

Range: 0 to 9

Group

Common commands

Syntax

```
*RCL
```

Arguments

Locations 1 to 9 are for arbitrary use. Location 0 is reserved for the recall of default instrument settings.

Examples

`*RCL 3` recalls instruments settings from location 3.

RPER

Sets or queries the PRBS period for pulse modulation. This value controls pulse modulation when the selected waveform is noise (see `PFNC`).

Group

Modulation commands

Syntax

```
RPER {t}[u]
RPER?
```

Related Commands PFNC

Arguments {t} is a floating point value. (If omitted, units default to seconds.)

Returns A floating point value.

Examples `RPER 1e-3` sets the bit period to 1 ms.

`RPER?` might return 0.001, indicating the bit period is 1 ms.

RPHS (No Query Form)

Sets the current phase of the carrier to 0°.

Group Signal synthesis commands

Syntax

```
RPHS
```

Arguments None

*RST (No Query Form)

Reset the instrument to default settings. This is equivalent to *RCL 0.

You can read more about default settings in the *TSG4100A Series RF Signal Generator User Manual* available for download at www.tektronix.com/manuals.

Group Common commands

Syntax

```
*RST
```

Related Commands *RCL

Arguments None

*SAV (No Query Form)

Save instrument settings to the specified location.

Conditions Range: 0 to 9

0 is reserved for current instrument settings.

Group Common commands

Syntax `*SAV`

Arguments Instrument setting location from 0 to 9.

Examples `*SAV3` saves the current setting to location 3.

SAVC? (Query Only)

Saves the user constellation stored in SRAM to the specified location in nonvolatile memory and returns an error code when complete.

Group Modulation commands

Syntax `SAVC? i`

Arguments i is an integer from 1 to 9.

Returns 0 means the save was successful.

Examples `SAVC? 3` saves the current constellation in SRAM to location 3 in nonvolatile memory.

SAVF? (Query Only)

Saves the user filter stored in SRAM to the specified location in nonvolatile memory and returns an error code when complete.

Group Modulation commands

Syntax `SAVF? i`

Arguments i is an integer from 1 to 9.

Returns 0 means the save was successful.

Examples `SAVF? 2` saves the current filter in SRAM to location 2 in nonvolatile memory.

SAVW? (Query Only)

Saves the user waveform stored in SRAM to the specified location in nonvolatile memory. When complete, returns an error code followed by the current free space in nonvolatile memory in 16-bit words.

Group Modulation commands

Syntax `SAVW? i`

Arguments i is an integer from 1 to 9.

Returns 0 (means the save was successful), 16-bit word

Examples `SAVW? 1` might return 0,1044480, indicating that the save was successful (0) and that there were 1,011,480 (1044480) words of free space available in nonvolatile memory for new waveforms.

SCAL

Sets or queries the digital scale factor for modulation waveforms. The digital scale factor enables the user to adjust the amplitude of the modulation waveform digitally. This factor applies to IQ modulation waveforms that vary the amplitude, such as ASK, PSK, and QAM, but not to constant amplitude waveforms, such as FM and CPM. As the scale factor is increased, the amplitude of the waveform is increased, but the likelihood that the waveform will be clipped is also increased. As the scale factor is decreased, the amplitude of the waveform will be decreased, but it will also be more susceptible to quantization noise.

This parameter should normally be left at 1.0.

Conditions Scale factor range: 0.2 to 3.0

Group Modulation commands

Syntax
```
SCAL {d}
SCAL?
```

Arguments d is a value between 0.2 and 3.0

Returns A value between 0.2 and 3.0

Examples `SCAL 2.0` sets the digital scale factor to 2.0.

`SCAL?` might return 1.0, indicating that the digital scale factor is set to 1.0.

*SRE

Sets or queries the Service Request Enable Register (SRER). Bits set in this register cause the instrument to generate a service request when the corresponding bit is set in the STB register. You can read more about the SRER and STB in the related sections. (See page 3-1.) (See page 3-2.)

Group Common commands

Syntax
```
*SRE {i}
```

Arguments {i} corresponds to the bits in the Status Byte register (STB), defined as follows:

Bit	Name	Meaning
0	INSB	An unmasked bit in the instrument status register (INSR) has been set.
1	Reserved	Not used.
2	Reserved	Not used.
3	Reserved	Not used.
4	MAV	The interface output buffer is non-empty.

Bit	Name	Meaning
5	ESB	An unmasked bit in the standard event status register (*ESR) has been set.
6	MSS	Master summary bit. Indicates that the instrument is requesting service because an unmasked bit in this register has been set.
7	Reserved	Not used.

Returns See Arguments.

STAG? (Query Only)

Queries whether the current constellation operates in staggered mode.

Group Modulation commands

Syntax `STAG?`

Returns 0 means staggered mode is disabled.

1 means staggered mode is enabled.

*STB? (Query Only)

Queries the standard IEEE 488.2 serial poll status byte.

Group Common commands

Syntax `*STB?`

Returns The bits in the STB register have the following meaning:

Bit	Name	Meaning
0	INSB	INSR summary bit
1	—	Reserved
2	—	Reserved
3	—	Reserved
4	MAV	Message available

Bit	Name	Meaning
5	ESB	ESR summary bit
6	MSS	Master summary bit
7	—	Reserved

INSB indicates that an enabled bit in INSR is set. MAV indicates that a message is available in the output queue. ESB indicates that an enabled bit in ESR is set. MSS reflects the fact that at least one of the summary enable bits is set and the instrument is requesting service.

Examples `*STB?` might return 113, indicating that the INSB, MAV, ESB, and MSS are set.

STYP

Sets or queries the modulation subtype.

Conditions Not all modulation subtypes are valid for each modulation type. Valid subtypes are shown in the following table.

Modulation type	Allowed subtypes
AM/ASK	0-5, 11
FM/FSK	0-5, 11
PM/PSK	0-5, 11-15
Pulse	0
QAM	3, 5-7, 9, 11
CPM	2-5
VSB	4-5

Group Modulation commands

Syntax

```
STYP {i}
STYP?
```

Arguments The parameter i may be set to one of the following values:

i	Modulation subtype
0	Analog (no constellation mapping)
1	Vector (no constellation mapping)
2	Default 1-bit constellation
3	Default 2-bit constellation

i	Modulation subtype
4	Default 3-bit constellation
5	Default 4-bit constellation
6	Default 5-bit constellation
7	Default 6-bit constellation
8	Default 7-bit constellation
9	Default 8-bit constellation
10	Default 9-bit constellation
11	User constellation
12	Factory OQPSK constellation
13	Factory DQPSK constellation
14	Factory π/4 DQPSK constellation
15	Factory 3π/8 8PSK constellation

Returns See Arguments.

Examples `STYP 3` selects the default 2-bit vector modulation.

`STYP?` might return 11, indicating that the modulation subtype is a user constellation.

SYMR

Sets or queries the symbol rate for digital waveforms.

Group Modulation commands

Syntax
```
SYMR {f}[u]
SYMR?
```

Related Commands See RATE to control modulation rates for functional waveforms in AM/FM/ΦM.

Arguments {i} is a floating point value. (If omitted, units [u] default to Hz.)

Returns A floating point value.

Examples `SYMR 270.833e3` sets the symbol rate to 270.833 kHz.

`SYMR10 kHz` set the symbol rate to 10 kHz.

SYMR? might return 100, indicating that the current rate in Hz.

SYMR? kHz might return 10, indicating that the current rate in kHz.

TDMA? (Query Only)

Queries the current TDMA configuration word.

Group Modulation commands

Syntax TDMA?

Returns An integer that signifies the configuration word in hex. The Time Domain Multiple Access (TDMA) configuration word is defined as follows:

Bit	31-18	17-16	15-12	11-8	7-1	0
Meaning	Reserved	Mux	Reserved	Ramp	Reserved	Enable

TDMA support is configured when the event marker files are downloaded through its own configuration word defined in the previous table.

Set bit 0 to enable TDMA functionality. Set Ramp to 1, 2, 4, or 8 for RF power ramp profiles that last the given number of symbol periods. Finally, set Mux to 1, 2, or 3 to associate the given marker with TDMA functionality. For simplicity the entire configuration word may be set to zero if TDMA is not used.

More information about TDMA waveforms is available in the *Event Markers and TDMA* section. (See page 3-4.)

Examples TDMA? might return 197121. In hexadecimal format, this number is 0x00030201. Referring to the definition of the TDMA configuration word, bit 0 indicates that TDMA is enabled; bits 11-8 indicate a ramp period of 2 symbols; and bits 17-16 indicate that the RF power is controlled by event marker #3.

TEMP? (Query Only)

Queries the current temperature of the RF output block in degrees C.

Group Status commands

Syntax TEMP?

Returns An integer indicating the temperature in degrees C.

Examples `TEMP?` might return 18, indicating the current temperature of the RF output block is 18 °C.

TIMB? (Query Only)

Queries the current timebase.

Group Status commands

Syntax `TIMB?`

Returns The following numbers identify the timebase:

Value	Meaning
0	Crystal timebase
1	OCXO timebase
2	Rubidium timebase
3	External timebase

*TRG (No Query Form)

This command initiates a trigger.

Group Common commands

Syntax `*TRG`

Arguments None.

*TST? (Query Only)

Runs the instrument self test.

Group Common commands

Syntax `*TST?`

Related Commands Use the LERR? command to determine the cause of a failure, if one occurs.

Returns 0 means the self test is successful.

17 is an error code that indicates the self test failed.

Examples `*TST?` might return 17, indicating that the self test failed. You can then use the LERR? command to determine the cause of the failure.

TYPE

Sets or queries the current modulation type. The modulation subtype, `STYP`, must also be specified to fully configure the modulation.

Group Modulation commands

Syntax
```
TYPE {i}
TYPE?
```

Related Commands STYP

Arguments The parameter i may be set to one of the following values:

i	Modulation type
0	AM/ASK
1	FM/FSK
2	ΦM/PSK
4	Pulse
7	QAM
8	CPM
9	VSB

Returns See Arguments.

Examples The following sequence would configure the modulation.

`TYPE 2` sets the modulation type to phase modulation.

STYP 3 sets the modulation subtype to the default 2-bit vector modulation.

UNLK? (Query Only)

Releases the instrument lock previously acquired by the LOCK? command.

Group Interface commands

Syntax UNLK?

Returns 0 means the lock is not released.

1 means the lock was released.

*WAI (No Query Form)

Sets the instrument to not process further commands until all prior commands including this one have completed.

Group Common commands

Syntax *WAI

Arguments None

Examples *WAI sets the instrument to wait for all prior commands to execute before continuing.

WAVF

Sets or queries the current user waveform.

Note that the user waveform is only active when the modulation function is set to user waveform (11).

Group Modulation commands

Syntax

```
WAVF {i}
WAVF?
```

Related Commands

MPRE

MFNC

PFNC

QFNC

Arguments

The parameter i identifies the desired user waveform. It may be one of the following values:

i	User waveform
0	User waveform stored in SRAM
1-9	User waveforms stored in nonvolatile memory
10	Audio clip
11	DECT frame of random data
12	TETRA frame of random data
13	GSM frame of random data
14	GSM EDGE frame of random data
15	W-CDMA mobile station frame of random data

Returns

See Arguments.

–1 means the current waveform is invalid. This might happen, for instance, if the waveform has been deleted.

10 and above refer to read-only waveforms loaded at the factory. These are typically configured as a consequence of modulation preset execution (`MPRE` command).

Examples

`WAVF 2` selects user waveform 2 for modulation.

WRTC (No Query Form)

Writes the given user constellation to SRAM.

Group

Modulation commands

Syntax

```
WRTC i,j,<arb data>
```

Arguments i is the bits/symbol for the constellation. It can be 1-9, 16, 32.

j = 0 means the constellation uses staggered IQ mode.

j = 1 means the constellation does not use staggered IQ mode.

<arb data> = a definite arbitrary block of binary data defining the constellation.

WRTE (No Query Form)

Writes the given event marker configuration to SRAM.

Group Modulation commands

Syntax `WRTE i,<arb data>`

Arguments i is the TDMA configuration word to apply the constellation.

<arb data> = a definite arbitrary block of binary data defining the event marker for the configuration.

WRTF (No Query Form)

Writes the given filter to SRAM.

Group Modulation commands

Syntax `WRTF i,<arb data>`

Arguments i is the offset to apply to each coefficient of the filter.

<arb data> = a definite arbitrary block of binary data defining the event marker for the configuration.

WRTW (No Query Form)

Write the given user waveform to SRAM.

Group Modulation commands

Syntax `WRTW i,j,<arb data>`

Arguments i is the bits/symbol for the constellation and whether it is an analog or vector waveform. It can be 1-9, 16, 32.

j indicates the number of bits in the waveform.

<arb data> = a definite arbitrary block of binary data defining the waveform.

XTRM (No Query Form)

Sets the interface terminator that is appended to each response to i, j, k. The default terminator is 13, 10, which is a carriage return followed by a line feed.

Group Interface commands

Syntax `XTRM i,j,k`

Arguments The default terminator is 13, 10, which is a carriage return followed by a line feed.

Status and Errors

Status and Events

The instrument reports on its status by means of the serial poll status byte and two event status registers: the standard event status (*ESR) and the instrument event status (INSR). These read-only registers record the occurrence of defined events inside the unit. If the event occurs, the corresponding bit is set to one. Bits in the status registers are latched. Once an event bit is set, subsequent state changes do not clear the bit. All bits are cleared when the registers are queried, with a *ESR?, for example. The bits are also cleared with the clear status command, *CLS. The bits are not cleared, however, with an instrument reset (*RST) or a device clear.

Status and event reporting

Each of the unit's event status registers has an associated enable register. The enable registers control the reporting of events in the serial poll status byte (*STB). If a bit in the event status register is set and its corresponding bit in the enable register is set, then the summary bit in the serial poll status byte (*STB) will be set. The enable registers are readable and writable. Reading the enable registers or clearing the status registers does not clear the enable registers. Bits in the enable registers must be set or cleared explicitly. To set bits in the enable registers, write an integer value equal to the binary weighted sum of the bits you wish to set.

The serial poll status byte (*STB) also has an associated enable register called the service request enable register (*SRE). This register functions in a similar manner to the other enable registers, except that it controls the setting of the master summary bit (bit 6) of the serial poll status byte. It also controls whether the unit will issue a request for service on the GPIB bus.

Status byte register

The Status Byte may be queried with the *STB? command. The Service Request Enable Register (*SRE) may be used to control when the instrument asserts the request-for-service line on the GPIB bus.

Table 3-1: Status byte register

Bit	Name	Meaning
0	INSB	An unmasked bit in the instrument status register (INSR) has been set.
1	Reserved	Not used.
2	Reserved	Not used.
3	Reserved	Not used.
4	MAV	The interface output buffer is non-empty.
5	ESB	An unmasked bit in the standard event status register (*ESR) has been set.

Table 3-1: Status byte register (cont.)

Bit	Name	Meaning
6	MSS	Master summary bit. Indicates that the instrument is requesting service because an unmasked bit in this register has been set.
7	Reserved	Not used.

Service Request Enable Register (SRER)

The SRER is made up of bits defined exactly the same as bits 0 through 7 in the SBR as shown in the previous table. This register is used by the user to determine what events will generate service requests.

The SRER bit 6 cannot be set. Also, the RQS is not maskable.

The generation of a service request with the GPIB interface involves changing the SRQ line to LOW and making a service request to the controller. The result is that a status byte for which an RQS has been set is returned in response to serial polling by the controller.

Use the *SRE command to set the bits of the SRER. Use the *SRE? query to read the contents of the SRER. Bit 6 must normally be set to 0.

Standard Event Status Register

Reports the power on/off state, command errors, and the running state. It consists of the Standard Event Status Register (SESR). This register is made up of the same bits defined in the following table. Use the *ESR? query to read the contents of the SESR. Use the *ESE() command to access the SESR.

Table 3-2: Standard event status register bit definition

Bit	Name	Description
7	PON	Power on. The unit has been power cycled.
6	Reserved	Not used.
5	CME	Command error. The parser detected a syntax error.
4	EXE	Execution error. A command failed to execute correctly because a parameter was invalid.
3	DDE	Device dependent error occurred.
2	QYE	Query error occurred.
1	Reserved	Not used.
0	OPC	Operation complete. All previous commands have completed. See command *OPC.

Instrument Status Register

The instrument status register may be queried with the INSR? command. The instrument status enable register (INSE) may be used to control the setting of the INSB summary bit in the serial poll status byte.

Table 3-3: Instrument status register bit definition

Bit	Name	Description
10–15	Reserved	Not used.
9	IQ_OVLD	An internal/external IQ modulation overload was detected.
8	MOD_OVLD	An internal/external modulation overload was detected.
7	Reserved	Not used.
6	RB_UNLK	An installed Rubidium timebase is unlocked.
5	NO_TIMEBASE	An installed optional timebase is not oscillating.
4	4GHZ_UNLK	The 4GHz PLL has come unlocked.
3	1GHZ_UNLK	The 1 GHz PLL has come unlocked.
2	19MHZ_UNLK	The 19 MHz PLL has come unlocked.
1	100MHZ_UNLK	The 100 MHz PLL has come unlocked.
0	20MHZ_UNLK	The 20 MHz PLL has come unlocked.

Event markers and TDMA

Event markers provide a means for synchronizing external equipment with a modulation waveform. A symbol clock and three event marker signals are available on the back panel of the instrument. The symbol clock output produces a square wave clock signal at the symbol rate. The rising edge of the clock aligns with the peak response of the symbol in the pulse shaping filter. At each rising edge of the symbol clock, the event markers may be programmed to pulse high or low.

Event markers are programmed with a sequence of up to 512, 32-bit configuration words, as follows.

Bit	31	30	29-0
Meaning	Repeat	Action	Offset

Bits 29 to 0 define a symbol offset. During initialization, a symbol counter is initialized to zero. At each rising edge of the symbol clock, the symbol counter is incremented. When the symbol count equals the offset given in the configuration word, the action in bit 30 is performed. If Action = 1, the event marker is forced high, otherwise it is forced low. If the symbol count does not equal the offset, no action is taken and the state of the marker is left unchanged. After each successful symbol count comparison, the next configuration word is loaded, until the repeat bit is set, at which point the symbol counter is reset and the first configuration word

reloaded. Each event marker has space for 512 event configuration words. For proper initialization, however, the first configuration word must have Offset = 0.

This simple system provides the flexibility to program markers that pulse high and low on every other symbol, or just once every million symbols. The symbol counters for each marker are independent, so we can have marker 1 pulsing high and low on every other symbol, but marker 2 independently pulsing high and low on every 10th symbol.

TDMA

These generators support Time Domain Multiple Access (TDMA) waveforms by associating an event marker with the RF power of output. When the marker is high, RF power is slowly ramped up to full power. When the marker is low, RF power is slowly ramped down and turned off. In the transition region the amplitude of the RF follows a raised cosine profile with a configurable period of 1, 2, 4, or 8 symbols. Once initiated, a ramp must complete before it may reversed, regardless of what the event marker it is associated with requests.

TDMA support is configured when the event marker files are downloaded through its own configuration word defined as follows.

Bit	31-18	17-16	15-12	11-8	7-1	0
Meaning	Reserved	Mux	Reserved	Ramp	Reserved	Enable

Set bit 0 to enable TDMA functionality. Set Ramp to 1, 2, 4, or 8 for RF power ramp profiles that last the given number of symbol periods. Finally, set Mux to 1, 2, or 3 to associate the given marker with TDMA functionality. For simplicity the entire configuration word may be set to zero if TDMA is not used.

Default event marker configuration

Marker configurations are always associated with a user waveform. When a user waveform is downloaded, a default marker configuration is automatically created for it. The newly created configuration replaces any other configuration that may have been active. The default marker configuration for a user waveform pulses Marker 1 high for one symbol period at the start of the waveform. Markers 2 and 3 are configured to stay low. TDMA is turned off. This default marker configuration is also used for PRBS and pattern waveforms, but in that case the configuration is fixed and cannot be modified.

Downloading event marker configurations

Event marker configurations are downloaded with the following command: WRTE i, <arb data>. Parameter i indicates the desired TDMA configuration word defined in the table in the *Event Markers and TDMA* section of this manual. <arb data> should be a definite arbitrary block with 6144 bytes of binary data. The binary data consists of three sequences of 512 event configuration words. The first sequence is for Marker 1, the next sequence is for Marker 2, and the last sequence is for Marker 3. The individual event configuration words are defined in the first table in the Event Markers and TDMA section. Each 32-bit configuration word must be formatted as big-endian for the transmission. (See page 3-3.)

Error messages and codes

Error codes with a negative value are SCPI standard error codes; errors with a positive value are unique to the generators.

The instrument contains an error buffer that may store up to 20 error codes associated with errors encountered during power-on self tests, command parsing, or command execution. An error message will appear when a remote command fails for any reason. The errors in the buffer may be read one by one by executing successive LERR? commands. The user may also view the errors from the front panel by pressing STATUS > Error Log to view the error count and individual errors. The errors are displayed in the order in which they occurred.

Execution errors

These error codes are returned when an error is detected while a command is being executed.

Table 3-4: Execution errors

Error code	Error and error message
0	No error No more errors left in the queue.
10	Illegal value A parameter was out of range.
11	Illegal mode The action is illegal in the current mode. This might happen, for instance, if the user tries to turn on IQ modulation with the MODL 1 command and the current frequency is below 400 MHz.
12	Not allowed The requested action is not allowed because the instrument is locked by another interface.
13	Recall failed The recall of instrument settings from nonvolatile storage failed. The instrument settings were invalid.
14	No clock option The requested action failed because the rear clock option is not installed.
15	No RF doubler option The requested action failed because the rear RF doubler option is not installed.
16	No IQ option The requested action failed because the rear IQ option is not installed.
17	Failed self test This value is returned by the *TST? command when the self test fails.

Query errors

These error codes are returned in response to errors that occur when attempting to read files.

Table 3-5: Query errors

Error code	Error message
30	Lost data Data was lost.
32	No Listener Listener is not found.

Device specific errors

These error codes are returned when an internal instrument error is detected. This type of error may indicate a hardware problem.

Table 3-6: Device specific errors

Error code	Error and error message
40	Failed ROM check The ROM checksum failed. The firmware code is likely corrupted.
42	Failed EEPROM check The test of EEPROM failed.
43	Failed FPGA check The test of the FPGA failed.
44	Failed SRAM check The test of the SRAM failed.
45	Failed GPIB check The test of GPIB communications failed.
46	Failed LF DDS check The test of the LF DDS communications failed.
47	Failed RF DDS check The test of the RF DDS communications failed.
48	Failed 20 MHz PLL The test of the 20 MHz PLL failed.
49	Failed 100 MHz PLL The test of the 100 MHz PLL failed.
50	Failed 19 MHz PLL The test of the 19 MHz PLL failed.
51	Failed 1 GHz PLL The test of the 1 GHz PLL failed.

Table 3-6: Device specific errors (cont.)

Error code	Error and error message
52	Failed 4 GHz PLL The test of the top octave PLL failed.
53	Failed DAC The test of the internal DACs failed.

File system errors

These error codes are returned in response to file system errors that occur when attempting to write or read files.

Table 3-7: File system errors

Error code	Error message
80	Out of memory Not enough memory to store the waveform.
81	File does not exist File does not exist.
82	File not open Cannot access a file that is not open.
83	Fille not writable File cannot be written.
84	File already exists Cannot create a file that already exists.
85	File corrupt File has been corrupted.
86	End of file Cannot read passed the end of the file.
87	File locked Cannot access a file because it is locked by another user.

Parsing errors

These error codes occur when commands are improperly parsed.

Table 3-8: Parsing errors

Error code	Error message
110	Illegal command The command syntax used was illegal. A command is normally a sequence of four letters, or a '*' followed by three letters.
111	Undefined command The specified command does not exist.
112	Illegal query The specified command does not permit queries
113	Illegal set The specified command can only be queried.
114	Null parameter The parser detected an empty parameter.
115	Extra parameters The parser detected more parameters than allowed by the command.
116	Missing parameters The parser detected missing parameters required by the command.
117	Parameter overflow The buffer for storing parameter values overflowed. This probably indicates a syntax error.
118	Invalid floating point number The parser expected a floating point number, but was unable to parse it.
120	Invalid integer The parser expected an integer, but was unable to parse it.
121	Integer overflow A parsed integer was too large to store correctly.
122	Invalid hexadecimal The parser expected hexadecimal characters but was unable to parse them.
126	Syntax error The parser detected a syntax error in the command.
127	Illegal units The units supplied with the command are not allowed.
128	Missing units The units required to execute the command were missing.

Communication errors

These error codes occur when communication is disrupted for some reason.

Table 3-9: Communication errors

Error code	Error message
170	Communication error A communication error was detected. This is reported if the hardware detects a framing, or parity error in the data stream.
171	Over run The input buffer of the remote interface overflowed. All data in both the input and output buffers will be flushed.

Other errors

These error codes occur for various reasons.

Table 3-10: Other errors

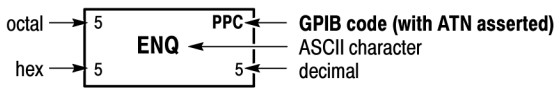
Error code	Error message
254	Too many errors The error buffer is full. Subsequent errors have been dropped.

Appendices

Appendix A: Character Charts

B7 B6 B5 / BITS B4 B3 B2 B1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		NUMBERS SYMBOLS		UPPER CASE		LOWER CASE	
0 0 0 0	0 NUL 0 0	20 DLE 10 16	40 LA0 SP 20 32	60 LA16 0 30 48	100 TA0 @ 40 64	120 TA16 P 50 80	140 SA0 ` 60 96	160 SA16 p 70 112
0 0 0 1	1 GTL SOH 1 1	21 LL0 DC1 11 17	41 LA1 ! 21 33	61 LA17 1 31 49	101 TA1 A 41 65	121 TA17 Q 51 81	141 SA1 a 61 97	161 SA17 q 71 113
0 0 1 0	2 STX 2 2	22 DC2 12 18	42 LA2 " 22 34	62 LA18 2 32 50	102 TA2 B 42 66	122 TA18 R 52 82	142 SA2 b 62 98	162 SA18 r 72 114
0 0 1 1	3 ETX 3 3	23 DC3 13 19	43 LA3 # 23 35	63 LA19 3 33 51	103 TA3 C 43 67	123 TA19 S 53 83	143 SA3 c 63 99	163 SA19 s 73 115
0 1 0 0	4 SDC EOT 4 4	24 DCL DC4 14 20	44 LA4 $ 24 36	64 LA20 4 34 52	104 TA4 D 44 68	124 TA20 T 54 84	144 SA4 d 64 100	164 SA20 t 74 116
0 1 0 1	5 PPC ENQ 5 5	25 PPU NAK 15 21	45 LA5 % 25 37	65 LA21 5 35 53	105 TA5 E 45 69	125 TA21 U 55 85	145 SA5 e 65 101	165 SA21 u 75 117
0 1 1 0	6 ACK 6 6	26 SYN 16 22	46 LA6 & 26 38	66 LA22 6 36 54	106 TA6 F 46 70	126 TA22 V 56 86	146 SA6 f 66 102	166 SA22 v 76 118
0 1 1 1	7 BEL 7 7	27 ETB 17 23	47 LA7 ' 27 39	67 LA23 7 37 55	107 TA7 G 47 71	127 TA23 W 57 87	147 SA7 g 67 103	167 SA23 w 77 119
1 0 0 0	10 GET BS 8 8	30 SPE CAN 18 24	50 LA8 (28 40	70 LA24 8 38 56	110 TA8 H 48 72	130 TA24 X 58 88	150 SA8 h 68 104	170 SA24 x 78 120
1 0 0 1	11 TCT HT 9 9	31 SPD EM 19 25	51 LA9) 29 41	71 LA25 9 39 57	111 TA9 I 49 73	131 TA25 Y 59 89	151 SA9 i 69 105	171 SA25 y 79 121
1 0 1 0	12 LF A 10	32 SUB 1A 26	52 LA10 * 2A 42	72 LA26 : 3A 58	112 TA10 J 4A 74	132 TA26 Z 5A 90	152 SA10 j 6A 106	172 SA26 z 7A 122
1 0 1 1	13 VT B 11	33 ESC 1B 27	53 LA11 + 2B 43	73 LA27 ; 3B 59	113 TA11 K 4B 75	133 TA27 [5B 91	153 SA11 k 6B 107	173 SA27 { 7B 123
1 1 0 0	14 FF C 12	34 FS 1C 28	54 LA12 , 2C 44	74 LA28 < 3C 60	114 TA12 L 4C 76	134 TA28 \ 5C 92	154 SA12 l 6C 108	174 SA28 ¦ 7C 124
1 1 0 1	15 CR D 13	35 GS 1D 29	55 LA13 – 2D 45	75 LA29 = 3D 61	115 TA13 M 4D 77	135 TA29] 5D 93	155 SA13 m 6D 109	175 SA29 } 7D 125
1 1 1 0	16 SO E 14	36 RS 1E 30	56 LA14 . 2E 46	76 LA30 > 3E 62	116 TA14 N 4E 78	136 TA30 ^ 5E 94	156 SA14 n 6E 110	176 SA30 ~ 7E 126
1 1 1 1	17 SI F 15	37 US 1F 31	57 LA15 / 2F 47	77 UNL ? 3F 63	117 TA15 O 4F 79	137 UNT – 5F 95	157 SA15 o 6F 111	177 RUBOUT (DEL) 7F 127
	ADDRESSED COMMANDS	UNIVERSAL COMMANDS	LISTEN ADDRESSES		TALK ADDRESSES		SECONDARY ADDRESSES OR COMMANDS	

KEY



Tektronix
REF: ANSI STD X3.4-1977
IEEE STD 488.1-1987
ISO STD 646-2973

Appendix B: GPIB Interface Specification

This appendix lists and describes the GPIB functions and messages the instrument can implement.

Interface Functions

The following table lists the GPIB interface functions this instrument implements. Each function is briefly described.

Table B-1: GPIB interface function implementation

Interface function	Implemented subset	Capability
Source Handshake (SH)	SH1	Complete
Acceptor Handshake (AH)	AH1	Complete
Talker (T)	T6	Basic Talker, Serial Poll Unaddress if my-listen-address (MLA) No Talk Only mode
Listener (L)	L4	Basic Listener Unaddress if my talk address (MTA) No Listen Only mode
Service Request (SR)	SR1	Complete
Remote/Local (RL)	RL0	None
Parallel Poll (PP)	PP0	None
Device Clear (DC)	DC1	Complete
Device Trigger (DT)	DT0	None
Controller (C)	C0	None
Electrical Interface	E2	Three-state driver

- Source Handshake (SH). Enables a talking device to support the coordination of data transfer. The SH function controls the initiation and termination of data byte transfers.
- Acceptor Handshake (AH). Enables a listening device to coordinate data reception. The AH function delays data transfer initiation or termination until the listening device is ready to receive the next data byte.
- Talker (T). Enables a device to send device-dependent data over the interface. This capability is available only when the device is addressed to talk, and uses a one-byte address.
- Listener (L). Enables a device to receive device-dependent data over the interface. This capability is available only when the device is addressed to listen, and uses a one-byte address.
- Service Request (SR). Enables a device to assert an SRQ (Service Request) line to notify the controller when it requires service.
- Remote/Local (RL). Enables a device to respond to both the GTL (Go To Local) and LLO (Local Lock Out) interface messages.
- Parallel Poll (PP). Enables a device to respond to the following interface messages: PPC, PPD, PPE, and PPU, as well as to send out a status message when the ATN (Attention) and EOI (End or Identify) lines are asserted simultaneously.
- Device Clear (DC). Enables a device to be cleared or initialized, either individually, or as part of a group of devices.
- Device Trigger (DT). Enables a device to respond to the GET (Group Execute Trigger) interface message when acting as a listener.
- Controller (C). Enables a device that has this capability to send its address, universal commands, and addressed commands to other devices over the interface.
- Electrical Interface (E). Identifies the electrical interface driver type. The notation E1 means the electrical interface uses open collector drivers, E2 means the electrical interface uses three-state drivers.

Interface Messages

Table B-2: Standard interface messages

Message	Type [1]	Implemented
Device Clear (DCL)	UC	Yes
Local Lockout (LLO)	UC	No
Serial Poll Disable (SPD)	UC	Yes
Serial Poll Enable (SPE)	UC	Yes

Table B-2: Standard interface messages (cont.)

Message	Type [1]	Implemented
Parallel Poll Unconfigure (PPU)	UC	No
Go To Local (GTL)	AC	Yes
Selected Device Clear (SDC)	AC	Yes
Group Execute Trigger (GET)	AC	No
Take Control (TCT)	AC	No
Parallel Poll Configure (PPC)	AC	No

1 UC: Universal command; AC: Address command

- Device Clear (DCL). Will clear (initialize) all devices on the bus that have a device clear function, whether or not the controller has addressed them.
- Local Lockout (LLO). Disables the return to local function.
- Serial Poll Disable (SPD). Changes all devices on the bus from the serial poll state to the normal operating state.
- Serial Poll Enable (SPE). Puts all bus devices that have a service request function into the serial poll enabled state. In this state, each device sends the controller its status byte, instead of its normal output, after the device receives its talk address on the data lines. This function may be used to determine which device sent a service request.
- Go To Local (GTL). Causes the listen-addressed device to switch from remote to local (front-panel) control.
- Select Device Clear (SDC). Clears or initializes all listen-addressed devices.
- Group Execute Trigger (GET). Triggers all applicable devices and causes them to initiate their programmed actions.
- Take Control (TCT). Allows the controller in charge to pass control of the bus to another controller on the bus.
- Parallel Poll Configure (PPC). Causes the listen-addressed device to respond to the secondary commands Parallel Poll Enable (PPE) and Parallel Poll Disable (PPD), which are placed on the bus following the PPC command. PPE enables a device with parallel poll capability to respond on a particular data line. PPD disables the device from responding to the parallel poll.

Index

R

S

T

U

W

X