



THS 710 & THS 720
TekScope™
Programmer Manual

070-9245-00

Copyright © Tektronix, Inc. 1995. All rights reserved. Licensed software products are owned by Tektronix or its suppliers and are protected by United States copyright laws and international treaty provisions.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, or subparagraphs (c)(1) and (2) of the Commercial Computer Software – Restricted Rights clause at FAR 52.227-19, as applicable.

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

Printed in the U.S.A.

Tektronix, Inc., P.O. Box 1000, Wilsonville, OR 97070-1000

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

WARRANTY

Tektronix warrants that this product will be free from defects in materials and workmanship for a period of three (3) years from the date of shipment. If any such product proves defective during this warranty period, Tektronix, at its option, either will repair the defective product without charge for parts and labor, or will provide a replacement in exchange for the defective product.

In order to obtain service under this warranty, Customer must notify Tektronix of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service. Customer shall be responsible for packaging and shipping the defective product to the service center designated by Tektronix, with shipping charges prepaid. Tektronix shall pay for the return of the product to Customer if the shipment is to a location within the country in which the Tektronix service center is located. Customer shall be responsible for paying all shipping charges, duties, taxes, and any other charges for products returned to any other locations.

This warranty shall not apply to any defect, failure or damage caused by improper use or improper or inadequate maintenance and care. Tektronix shall not be obligated to furnish service under this warranty a) to repair damage resulting from attempts by personnel other than Tektronix representatives to install, repair or service the product; b) to repair damage resulting from improper use or connection to incompatible equipment; or c) to service a product that has been modified or integrated with other products when the effect of such modification or integration increases the time or difficulty of servicing the product.

THIS WARRANTY IS GIVEN BY TEKTRONIX WITH RESPECT TO THIS PRODUCT IN LIEU OF ANY OTHER WARRANTIES, EXPRESSED OR IMPLIED. TEKTRONIX AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TEKTRONIX' RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY. TEKTRONIX AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER TEKTRONIX OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Table of Contents

General Safety Summary	vii
Preface	ix
Related Manuals	ix
Conventions	x

Getting Started

Setting Up Remote Communications	1-1
Programming Model	1-10
Tutorial	1-10

Syntax and Commands

Command Syntax	2-1
Command and Query Structure	2-1
Command Entry	2-4
Constructed Mnemonics	2-7
Argument Types	2-8
Command Groups	2-11
Oscilloscope Commands	2-11
DMM Commands	2-32
Command Descriptions	2-39

Status and Events

Registers	3-1
Queues	3-6
Event Handling Sequence	3-7
Synchronization Methods	3-9
Messages	3-13

Examples

Programming Examples	4-1
Compiling the Example programs	4-2

Appendices

Appendix A: ASCII Code Chart	A-1
Appendix B: Reserved Words	B-1
Appendix C: Factory Setup	C-1

Glossary and Index

Glossary

Index

List of Figures

Figure 1–1: RS-232 Connector Location and Pin Assignments	1–2
Figure 2–1: Block Argument Example	2–10
Figure 3–1: The Standard Event Status Register (SESR) ..	3–2
Figure 3–2: The Status Byte Register (SBR)	3–3
Figure 3–3: The Device Event Status Enable Register (DESER)	3–4
Figure 3–4: The Event Status Enable Register (ESER)	3–5
Figure 3–5: The Service Request Enable Register (SRER)	3–5
Figure 3–6: Status and Event Handling Process	3–8
Figure 3–7: Command Processing Without Using Synchronization	3–10
Figure 3–8: Processing Sequence With Synchronization ...	3–10
Figure 4–1: Equipment Needed to Run the RS-232 Example Programs	4–1

List of Tables

Table 1–1: Connecting to Common Devices	1–3
Table 1–2: RS-232 Default Settings	1–5
Table 1–3: RS-232 Troubleshooting	1–8
Table 1–4: Comparison of Programming and Front-Panel Models	1–10
Table 2–1: BNF Symbols and Meanings	2–1
Table 2–2: Command Message Elements	2–2
Table 2–3: Comparison of Header Off and On Responses .	2–4
Table 2–4: Acquisition Commands	2–11
Table 2–5: Alias Commands	2–12
Table 2–6: Calibration and Diagnostic Commands	2–12
Table 2–7: Cursor Commands	2–13
Table 2–8: Display Commands	2–14
Table 2–9: Hard Copy Commands	2–15
Table 2–10: Horizontal Commands	2–16
Table 2–11: Oscilloscope Measurement Commands	2–17
Table 2–12: Miscellaneous Oscilloscope Commands	2–19
Table 2–13: RS-232 Commands	2–21
Table 2–14: Oscilloscope Save and Recall Commands	2–22
Table 2–15: Status and Error Commands	2–22
Table 2–16: Trigger Commands	2–23
Table 2–17: Vertical Commands	2–25
Table 2–18: Waveform Commands	2–30
Table 2–19: Miscellaneous DMM Commands	2–32
Table 2–20: DMM Calibration Commands	2–33
Table 2–21: DMM Function Range Commands	2–33
Table 2–22: DMM Measurement Commands	2–33

Table 2–23: DMM Probe Commands	2–34
Table 2–24: DMM Beeper Commands	2–35
Table 2–25: DMM Logger Commands	2–36
Table 2–26: DMM Save and Recall Commands	2–37
Table 2–27: Commands that Affect BUSY? Response	2–54
Table 2–28: DATA and WFMPre Parameter Settings	2–85
Table 2–29: XY Format Pairs	2–97
Table 2–30: Commands that Generate an Operation Complete Message	2–190
Table 2–31: Additional WFMPre Commands	2–248
Table 3–1: SESR Bit Functions	3–2
Table 3–2: SBR Bit Functions	3–3
Table 3–3: No Event Messages	3–14
Table 3–4: Command Error Messages – CME Bit 5	3–14
Table 3–5: Execution Error Messages – EXE Bit 4	3–15
Table 3–6: Device Error Messages – DDE Bit 3	3–18
Table 3–7: System Event Messages	3–19
Table 3–8: Execution Warning Messages – EXE Bit 4	3–19
Table 3–9: Internal Warning Messages	3–20

Table of Contents



General Safety Summary

Review the following safety precautions to avoid injury and prevent damage to this product or any products connected to it.

Only qualified personnel should perform service procedures.

Injury Precautions

Use Proper Power Cord. To avoid fire hazard, use only the power cord specified for this product.

Avoid Electric Overload. To avoid injury or fire hazard, do not apply potential to any input, including the common inputs, that varies from ground by more than the maximum rating for that input.

Avoid Electric Shock. To avoid injury or loss of life, do not connect or disconnect probes or test leads while they are connected to a voltage source.

Do Not Operate Without Covers. To avoid electric shock or fire hazard, do not operate this product with covers or panels removed.

Do Not Operate in Wet/Damp Conditions. To avoid electric shock, do not operate this product in wet or damp conditions.

Do Not Operate in an Explosive Atmosphere. To avoid injury or fire hazard, do not operate this product in an explosive atmosphere.

Product Damage Precautions

Use Proper Power Source. Do not operate this product from a power source that applies more than the voltage specified.

Do Not Operate With Suspected Failures. If you suspect there is damage to this product, have it inspected by qualified service personnel.

Safety Terms and Symbols

Terms in This Manual. These terms may appear in this manual:



WARNING. *Warning statements identify conditions or practices that could result in injury or loss of life.*



CAUTION. *Caution statements identify conditions or practices that could result in damage to this product or other property.*

Terms on the Product. These terms may appear on the product:

DANGER indicates an injury hazard immediately accessible as you read the marking.

WARNING indicates an injury hazard not immediately accessible as you read the marking.

CAUTION indicates a hazard to property including the product.

Symbols on the Product. These symbols may appear on the product:



DANGER
High Voltage



Protective Ground
(Earth) Terminal



ATTENTION
Refer to Manual



Double
Insulated

Certifications and Compliances

CSA Certified AC Adapter. CSA Certification includes the AC adapters appropriate for use in the North America power network. All other AC adapters supplied are approved for the country of use.

Compliances. Consult the product specifications for Overvoltage Category and Safety Class.

Preface

This is the Programmer Manual for the THS 700 Series TekScope instruments. This manual provides information about operating your instrument using the RS-232 interface.

Related Manuals




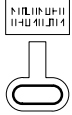


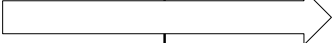


Additional documentation for the instrument is contained in the related manuals listed below.

Language	User Manual Part Number	Reference Part Number	Service Manual Part Number
English	070-9247-XX	070-9257-XX	070-9246-XX
French	070-9248-XX	070-9275-XX	
German	070-9249-XX	070-9276-XX	
Italian	070-9250-XX	070-9277-XX	
Korean	070-9251-XX	070-9278-XX	
Portuguese	070-9252-XX	070-9279-XX	
Spanish	070-9253-XX	070-9280-XX	
Simple Chinese	070-9254-XX	070-9281-XX	
Standard Chinese	070-9255-XX	070-9256-XX	
Japanese	070-9258-XX		

Conventions

TekScope instrument setups are shown in tables. The *Getting Started* chapter uses a table to show the complete contents of a menu.

The header of each table contains icons that represent the controls and menu items used to set up the instrument. To make a specific setup, read the table from left to right and then from top to bottom as shown below. The table contains the symbol “—” if no action is required.

 METER  SCOPE				TOGGLE 
1. Choose scope mode or meter mode.	2. Press this button on the front panel.	3. Press this bezel button.	4. Press the bezel button again until this selection is highlighted.	5. Use the +/- rocker to set the value for a parameter.
	6. 	7. 	—	
	8. 	9.	10.	



Getting Started

Getting Started

This chapter covers the following topics.

- *Setting Up Remote Communications* describes setting up for remote control including connecting the TekScope instrument and setting the appropriate front panel controls.
- *Programming Model* describes how the programming interface differs from the front-panel user interface.
- *Tutorial* includes a simple programming example.

For general information about operating the instrument, refer to the *THS 710 & THS 720 User Manual*.

Setting Up Remote Communications

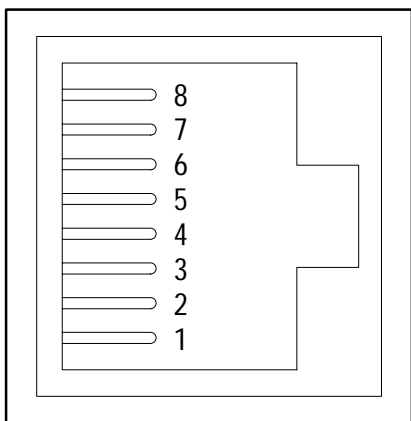
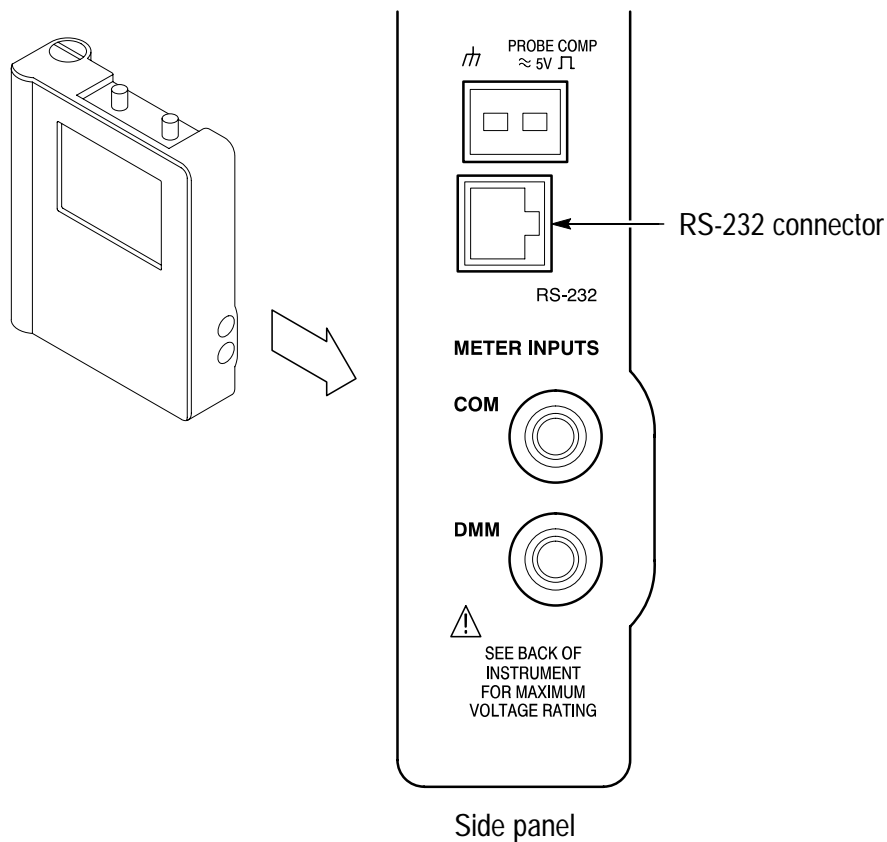
The THS 700 Series TekScope instrument has an RS-232 connector located on its right side, as shown in Figure 1–1. This connector is an RJ-45 connector. You can use adapters to convert to either 9-pin or 25-pin RS-232 connectors which conform to ANSI/EIA/TIA Standard 574–1990.

Connecting to an RS-232 Device

The RS-232 interface provides a point-to-point connection between two items of equipment such as a computer or terminal and the TekScope instrument. This section tells how to connect and set up the instrument for communication over the RS-232 interface.

Table 1–1 shows how to use parts in the RS-232 Cable Set (standard accessory) with other adapters to connect the instrument to common devices. You can purchase the other adapters from most computer supply dealers.

Two types of RS-232 devices are defined: Data Terminal Equipment (DTE) and Data Communications Equipment (DCE). The TekScope instrument is a DTE device.



Pin assignments

- 1 Data Carrier Detect (DCD)
- 2 Data Set Ready (DSR)
- 3 Data Terminal Ready (DTR)
- 4 Signal Ground (GND)
- 5 Receive Data (RxD)
- 6 Transmit Data (TxD)
- 7 Clear to Send (CTS)
- 8 Request to Send (RTS)






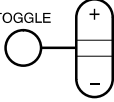
Figure 1-1: RS-232 Connector Location and Pin Assignments

Table 1-1: Connecting to Common Devices

To connect to device type	Use these parts from the RS-232 Cable Set (standard accessory)	With these additional adapters
HC 411 or DPU 411-II printer	RS-232 cable	RJ-45 to 25-pin male adapter (Tektronix part number 103-0334-XX, supplied with HC 411)
Most other serial printers	RS-232 cable and RJ-45 to 9-pin female null modem adapter	9-pin male to 25-pin male adapter (Radio Shack part number 26-1388)
Sun workstations		
PC/AT or laptop with 9-pin male connector	RS-232 cable and RJ-45 to 9-pin female null modem adapter	None required
Old-style PC with 25-pin male connector	RS-232 cable and RJ-45 to 9-pin female null modem adapter	9-pin male to 25-pin male adapter (Radio Shack part number 26-1388), followed by 25-pin female to 25-pin female gender adapter (Radio Shack part number 26-1495)

Configuring the TekScope Instrument

You can configure the TekScope instrument RS-232 port while in scope or meter mode. The contents of the RS-232 System branch of the Utility menu are summarized in the table below.

 METER  SCOPE				
SCOPE or METER	UTILITY	System	RS-232	—
		Baud Rate	—	<i>Choose rate</i>
		Flagging	Hard Flagging	On
			Soft Flagging	Off
		Miscellaneous	EOL	CR LF CR/LF LF/CR
			Parity	None Even Odd
			Stop Bits	1 2
			Delay	<i>Set delay</i>
Set RS232 Parameters to Defaults	—	—		

After these parameters are set, the RS-232 interface is ready to operate:

- **Baud Rate** — sets the data transmission rate in the range from 300 to 38,400 baud.
- **Hard Flagging** — sets hard flagging (RTS/CTS) on or off. Hard flagging is the preferred method of controlling the flow of data between devices. Use hard flagging for binary data transfers and instrument control.

- **Soft Flagging** — sets soft flagging (XON/XOFF) on or off. Use soft flagging only if it is required by your printer.
- **EOL** — sets the end of line terminator sent by the instrument. You can set CR, LF, LF/CR, or CR/LF.
- **Parity** — adds an error check bit (ninth bit) to each character. You can set the error bit for either None, Even, or Odd parity. When the parity setting is odd or even, the instrument generates the selected parity on output and checks incoming data for the selected parity. When the parity setting is none, there is no parity bit.
- **Stop Bits** — sets the number of stop bits sent after each character. You can set 1 or 2 stop bits.
- **Delay** — sets the delay time before responding to a query. You can set times from 0 s to 60 s in 100 ms increments.
- **Set RS-232 Parameters to Defaults** — sets RS-232 parameters to the default values listed in Table 1–2.

Table 1–2: RS-232 Default Settings

RS-232 Parameter	Default Setting
Baud Rate	9600
Hard Flagging	On
Soft Flagging	Off
EOL	LF
Parity	None
Stop Bits	1
Delay	0 s

RS-232 Conventions

You should be aware of the processing conventions that are specific to the RS-232 interface. These conventions pertain to the following tasks:

- Transferring binary data
- Processing break signals
- Reporting RS-232 I/O errors
- Checking Command Status

Transferring Binary Data. When transferring binary data to the TekScope instrument via the RS-232 port, note the following points:

- Using RTS/CTS (hard) flagging guarantees no data loss.
- All eight bits of binary data contain meaningful information. To ensure that all eight bits are received or transmitted, an RS-232 device which is connected to the instrument must be configured to receive and transmit eight-bit characters (set the RS-232 word length to eight bits).

Processing Break Signals. When the TekScope instrument senses a break signal on the RS-232 port, it returns DCL followed by the end of line terminator. Break signals do not change instrument settings or stored data and do not interrupt front-panel operation or nonprogrammable functions.

If a break signal is sent in the middle of a character stream, several characters immediately preceding or following the break may be lost. The controller should wait until it receives the DCL and end of line terminator string before sending more characters.

Reporting RS-232 I/O Errors. Errors are reported when there is a problem with parity, framing, or input/output buffer overruns. To report errors, the TekScope instrument posts an event code (refer to *Status and Events*). When an error occurs, the instrument discards all input and output and waits for a new command. A count of these errors since last power on is included in the error log (in the Diag System of the Utility menu).

You can use the RS232 Line Snapshot entry of the error log to help establish an RS-232 connection. The snapshot reports whether the instrument is waiting to receive a control-Q (yes/no), the state of the hardware CTS line (high/low), and whether characters have been received (yes/no). Use the following statements to help you interpret the status reported in the error log:

- If soft flagging is on and if Waiting For ^Q is Yes, the instrument must receive an XON character before it will transmit any more data.
- If hard flagging is on and CTS is Low, the instrument will not transmit any data.
- If hard flagging is off, you should ignore the value of CTS since the instrument ignores it.
- If Chars Rcvd is Yes, the instrument has received at least one character since the last power-on.

The RS232 Errors line of the error log lists the number of parity, framing, and overrun errors since the last power-on.

Checking Command Status. If you want to check the status of each command sent, you can append a *STB? query after every command and read the response string.

RS-232 Troubleshooting

If the TekScope instrument and the personal computer or printer have trouble communicating, use the following steps to correct the problem:

1. Verify that you are using the correct RS-232 cable and adapters. Determine whether your configuration requires a null-modem connection (where transmit/receive and control lines are switched) or a straight-through RS-232 connection. Refer to Table 1–1 for information about RS-232 connections.
2. Verify that the RS-232 cable and adapters are firmly connected to both the TekScope instrument and the correct port on your personal computer or printer. Verify that your printer or the program on the personal computer is using the correct port. Try your program or printer again.

3. Verify that the TekScope instrument settings match the settings used by your printer or the program on your personal computer. Start by choosing Set RS-232 Parameters to Defaults (in the RS-232 System of the Utility menu). Then, change only those menu items that you know need to be changed, such as the baud rate. Try your printer or computer program again.
4. If you are trying to control the instrument using a personal computer or other computer, look at the diagnostic error log and examine the RS232 Line Snapshot and the RS232 Errors. The RS232 Line Snapshot and the RS232 Errors will not change while you are viewing them. They are reset when the power is turned on. Use Table 1–3 to troubleshoot your setup.

Table 1–3: RS-232 Troubleshooting

Symptom	Possible Causes
Your personal computer program tried to send characters to the TekScope instrument, but the error log displays Chars Rcvd: No.	Your RS-232 cable may be wired as a modem instead of a null modem. If you are attempting to use a telephone modem, the cable may be wired as a null modem instead of a modem.
The instrument error log displays Framing errors.	<p>There is a baud rate mismatch between the instrument and the personal computer.</p> <p>There is a data bits mismatch between the instrument and the personal computer (The instrument expects 8-bit data).</p> <p>There is a parity mismatch between the instrument and the personal computer.</p>
The instrument error log displays Parity errors.	There is a parity mismatch between the instrument and the personal computer.

Table 1–3: RS-232 Troubleshooting (Cont.)

Symptom	Possible Causes
The instrument error log displays Overrun errors.	Flagging is not being used correctly by the instrument or the personal computer: they are using different types of flagging, or they are using soft flagging for instrument control.
Transmissions are incomplete, or the instrument does not process all commands from the personal computer.	Flagging is not being used correctly by the instrument or the personal computer: they are using different types of flagging, or they are using soft flagging for instrument control. There is an EOL terminator mismatch between the instrument and the personal computer.
The instrument error log displays Waiting for ^Q: Yes.	The instrument is using soft flagging. Use hard flagging instead unless your printer requires soft flagging.
Soft flagging is being used, and transmissions stop.	Verify that both the personal computer and the instrument are not sending binary data. Binary data may contain ^S characters which cause transmissions to stop. Use hard flagging instead unless your printer requires soft flagging.
The instrument error log displays CTS: Low, and the instrument is using hard flagging.	Verify that the RS-232 cable is the recommended cable. Some cables may be wired without the CTS or RTS lines which are used by hard flagging. Verify that the personal computer program is using CTS/RTS hard flagging.
After the personal computer program sends a BREAK, the first message fails.	Verify that the personal computer program is waiting for and reading the DCL and end of line terminator response sent by the instrument.

Programming Model

Table 1–4 describes how the model for programming the TekScope instrument differs from controlling the instrument using the front panel.

Table 1–4: Comparison of Programming and Front-Panel Models

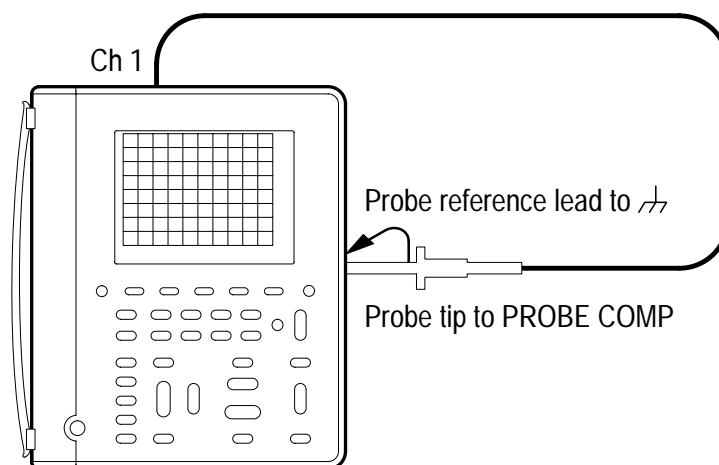
Programming Model	Front-Panel Model
You can access all controls at all times.	You can only change the controls that are accessible in the current menu or instrument mode.
You have more control over some features than is available through the front panel.	You have less control over some features than is available through the programming interface.

Tutorial

This tutorial contains one simple example of operating the TekScope instrument through the programming interface. This example verifies communication with the instrument, acquires a signal, and then takes a frequency measurement. Refer to the chapter *Examples* for a description of the diskette, included with this manual, that contains additional programming examples.

The tutorial assumes you have already attached a PC to the instrument and that the PC is executing a terminal-emulator program (such as a Microsoft Windows Terminal, for example). Be sure to set the communication settings in the terminal-emulator program to match those of the TekScope instrument.

1. Connect the oscilloscope probe to the channel 1 input BNC. Attach the probe tip and reference lead to the PROBE COMP connectors on the right side of the TekScope instrument. The PROBE COMP signal is a square wave with a frequency of ≈ 1.2 kHz.



2. Press the **ON/OFF** button to turn on the TekScope instrument.
3. After a few seconds, you should see a window with the message Power-On self check PASSED. Press the **CLEAR MENU** button to begin operation.
4. Send the query ID?. The instrument responds with its identification string ID TEK/THS 710,CF:91.1CT,FV:v1.00 (or similar).
5. Send the command FACTory to reset the instrument to the factory default state. The factory default state is described in *Appendix C: Factory Setup*.
6. Send the command AUTORange ON to turn on the autorange function.
7. Send the command MEASurement:IMMed:SOURCE CH1 to select measurements on channel 1.
8. Send the command MEASurement:IMMed:TYPe FREquency to set up the frequency measurement.

9. Send the query `MEASUREMENT:IMMED:VALUE?` to request the measurement result. The instrument will respond with a result similar to `1.199064819335938E3`, which is a frequency measurement of the `PROBE COMP` signal.

This completes the tutorial.



Syntax and Commands

Command Syntax

You can control the TekScope instrument through the RS-232 interface using a large group of commands and queries. This section describes the syntax these commands and queries use and the conventions the instrument uses to process them. The commands and queries themselves are listed in the *Command Descriptions* section.

You transmit commands to the instrument using the enhanced American Standard Code for Information Interchange (ASCII) character encoding. *Appendix A* contains a chart of the ASCII character set.

This manual uses Backus-Naur Form (BNF) notation and syntax diagrams to describe commands and queries. Table 2–1 lists the BNF symbols that this manual uses.

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
. . .	Previous element(s) may be repeated
()	Comment

Command and Query Structure

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

Most commands have both a set form and a query form. The query form of the command is the same as the set form except if it ends with a question mark. 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 are set only and some are query only.

A command message is a command or query name, followed by any information the instrument needs to execute the command or query. Command messages consist of five different element types, defined in Table 2–2.

Table 2–2: Command Message Elements

Symbol	Meaning
<Header>	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. The beginning colon can never be used with command headers beginning with a star (*).
<Mnemonic>	A header subfunction. Some command headers have only one mnemonic. If a command header has multiple mnemonics, they are always separated from each other by a colon (:) character.
<Argument>	A quantity, quality, restriction, or limit associated with the header. Not all commands have an argument, while other commands have multiple arguments. Arguments are separated from the header by a <Space>. Arguments are separated from each other by a <Comma>.
<Comma>	A single comma between arguments of multiple-argument commands. It may optionally have white space characters before and after the comma.
<Space>	A white space character between command header and argument. It may optionally consist of multiple white space characters.

Commands

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

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

A command header is made up 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 of the previous one. Commands at a higher level in the tree may affect those at a lower level. The leading colon (:) always returns you to the base of the command tree.

Queries

Queries cause the instrument to return information about its status or settings. 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. For example, `DISplay:CONtrast?` returns only the display contrast setting while `DISplay?` returns all the display settings.

Headers in Query Responses

You can control whether the instrument returns headers as part of the query response. Use the `HEADER` command to control this feature. If header is on, the instrument returns command headers as part of the query and formats the query response as a valid set command. When header is off, the instrument sends back only the values in the response. This format can make it easier to parse and extract the information from the response. Table 2–3 shows the difference in responses.

Table 2-3: Comparison of Header Off and On Responses

Query	Header Off Response	Header On Response
CURSor:VBArS: DELTA?	1.064E-03	:CURSor:VBArS: DELTA 1.064E-03
ACQuire:NUMAVg?	16	:ACQUIRE:NUMAVG 16

Output Queue

To clear the Output Queue and reset the instrument to accept a new command or query, use the RS-232 break signal.

Command Entry

Follow these general rules 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 null (00), tab (09), and space (20).
- You can enter a command that consists of any combination of white space characters and line feeds; the instrument will ignore this command.

Abbreviating Commands

You can abbreviate many instrument commands. These abbreviations are shown in capitals in the command listing in the *Command Groups* section on page 2-11 and *Command Descriptions* section on page 2-39. For example, the command ACQuire:NUMAVg can be entered simply as ACQ:NUMA or acq:numa.

If you use the HEADer command to have command headers included as part of query responses, you can also control whether the returned headers are abbreviated or are full-length using the VERBoSe command.

Concatenating Commands

You can concatenate any combination of set commands and queries using a semicolon (;). The instrument executes concatenated

commands in the order received. When concatenating commands and queries you must follow these rules:

- Completely different headers must be separated by both a semicolon and by the beginning colon on all commands but the first. For example, the commands TRIGger:MAIn:MODE NORMAl and ACQuire:NUMAVg 8 can be concatenated into a single command:

```
TRIGger:MAIn:MODE NORMAl;:ACQuire:NUMAVg 8
```

- 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, the commands ACQuire:MODE ENVeloPe and ACQuire:NUMAVg 4 could be concatenated into a single command:

```
ACQuire:MODE ENVeloPe; NUMAVg 4
```

The longer version works equally well:

```
ACQuire:MODE ENVeloPe;:ACQuire:NUMAVg 4
```

- Never precede a star (*) command with a colon:

```
ACQuire:MODE ENVeloPe;*TRG
```

The instrument processes commands that follow the star command as if the star command was not there so this concatenation with the last command in abbreviated form is legal:

```
ACQuire:MODE ENVeloPe;*TRG;NUMAVg 4
```

- When you concatenate queries, the instrument concatenates responses to all the queries into a single response message. For example, if the RS-232 parameters are set to their default values, the concatenated query:

```
RS232:HARDFlagging?;SOFTFlagging?
```

will return 1;0.

- You may concatenate set commands and queries in the same message. For example:

```
ACQuire:MODE NORMAl;NUMAVg?;STATE?
```

is a valid message that sets the acquisition mode to normal, queries the number of acquisitions for averaging, and then queries the acquisition state. The instrument executes concatenated commands and queries in the order it receives them.

Here are some invalid concatenation examples:

- TRIGger:MAIn:MODE NORMAl;ACQuire:NUMAVg 16
(missing colon before ACQuire)
- TRIGger:MAIn:MODE NORMAl;:EDGE:COUPling DC
(extra colon before EDGE)
- TRIGger:MAIn:MODE NORMAl;:*TRG
(extra colon before a star (*) command)

Message Terminator

This manual uses <EOM> (End of message) to represent a message terminator.

RS-232 End of Message Terminators. May be a CR (carriage return), LF (line feed), CRLF (carriage return followed by a line feed), or LFCR (line feed followed by a carriage return). When receiving, the instrument accepts all four combinations as valid input message terminators regardless of the currently selected terminator. When a combination of multiple characters is selected (CRLF or LFCR), the instrument interprets the first character as the terminator and the second character as a null command.

Constructed Mnemonics

Cursor Position Mnemonics

When the instrument displays cursors, commands may specify which cursor of the pair to use.

Symbol	Meaning
POSITION<x>	A cursor selector; <x> is either 1 or 2.

Measurement Specifier Mnemonics

Commands can specify which measurement to set or query as a mnemonic in the header. In scope mode, the instrument can display up to four automated measurements. The displayed measurements are specified in this way:

Symbol	Meaning
MEAS<x>	A measurement specifier; <x> is either 1, 2, 3, or 4.

Channel Mnemonics

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

Symbol	Meaning
CH<x>	A channel specifier; <x> is either 1 or 2.

Math Waveform Mnemonics

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

Symbol	Meaning
MATH<x>	A math waveform specifier; <x> is 1.

Reference Waveform Mnemonics

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

Symbol	Meaning
REF<x>	A reference waveform specifier; <x> is either A or B. (Specifier values 1 and 2 are also allowed for backward compatibility with other instruments.)

Waveform Mnemonics

In some commands you can specify a waveform without regard to its type: channel waveform, math waveform, or reference waveform:

Symbol	Meaning
<wfm>	Can be CH<x>, MATH, MATH1, or REF<x>

Argument Types

The argument of a command may be in one of several forms. The individual descriptions of each command tell which argument types to use with that command.

Numeric Arguments

Many instrument commands require numeric arguments. The syntax shows the format that the instrument returns in response to a query. This response is also the preferred format when sending the command to the instrument though it will accept any of the formats. This manual represents these arguments as follows:

Symbol	Meaning
<NR1>	Signed integer value
<NR2>	Floating point value without an exponent
<NR3>	Floating point value with an exponent

The instrument will automatically force most numeric arguments to a valid setting, either by rounding or truncating, when you input an invalid number. Exceptions are noted in the command descriptions.

Quoted String Arguments

Some commands accept or return data in the form of a quoted string, which is simply a group of ASCII characters enclosed by single quotes (') or double quotes ("). For example:

```
"this is a quoted string"
```

Symbol	Meaning
<QString>	Quoted string of ASCII text

Follow these rules when you use quoted strings:

1. A quoted string can include any character defined in the 7-bit ASCII character set. (See *Appendix A*).
2. Use the same type of quote character to open and close the string:


```
"this is a valid string"
```
3. You can mix quotation marks within a string as long as you follow the previous rule:


```
"this is an 'acceptable' string"
```
4. You can include a quote character within a string simply by repeating the quote. For example,


```
"here is a "" mark"
```
5. Strings can have upper or lower case characters.
6. You cannot terminate a quoted string with the END message before the closing delimiter.
7. A carriage return or line feed embedded in a quoted string does not terminate the string, but is treated as just another character in the string.
8. The maximum length of a quoted string returned from a query is 1000 characters.

Here are some examples of invalid strings:

"Invalid string argument'
 (quotes are not of the same type)

"test<LF>"
 (termination character is embedded in the string)

Block Arguments

Several instrument commands use a block argument form:

Symbol	Meaning
<NZDig>	A non-zero digit character, in the range 1–9
<Dig>	A digit character, in the range 0–9
<DChar>	A character with the hex equivalent of 00 through FF hexadecimal (0 through 255 decimal)
<Block>	A block of data bytes, defined as: <Block> ::= #<NZDig><Dig>[<Dig>...] [<DChar>...]

<NZDig> specifies the number of <Dig> elements that follow. Taken together, the <Dig> elements form a decimal integer that specifies how many <DChar> elements follow. Figure 2–1 shows a block argument example.

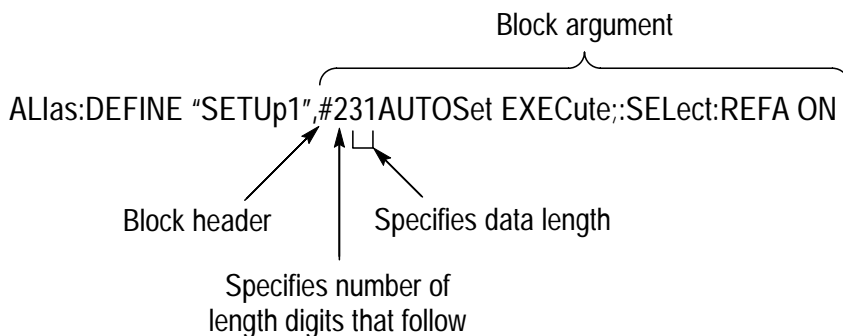


Figure 2–1: Block Argument Example

Command Groups

TekScope instrument commands are divided into two categories, Oscilloscope Commands and DMM Commands, which correspond to the two primary modes of instrument operation. Within these two categories, this section lists the commands organized by functional group. The *Command Descriptions* section on page 2–39 lists all commands alphabetically.

The instrument RS-232 interface conforms to Tektronix standard codes and formats except where noted.

Oscilloscope Commands

The following command groups cover general instrument functions and those specific to scope-mode operation.

Acquisition Commands

Acquisition commands affect the acquisition of waveforms. These commands control mode, averaging, enveloping, and single-waveform acquisition.

Table 2–4: Acquisition Commands

Header	Description
ACQUIRE?	Return acquisition parameters
ACQUIRE:MODE	Query/set acquisition mode
ACQUIRE:NUMACQ?	Return # of acquisitions obtained
ACQUIRE:NUMAVG	Query/set number of acquisitions for average
ACQUIRE:NUMENV	Query/set number of acquisitions for envelope
ACQUIRE:STATE	Start or stop acquisition system
ACQUIRE:STOPAFTER	Query/set acquisition control

Alias Commands

Alias commands let you define your own commands as a sequence of standard commands. This feature is useful when you use the same commands each time you perform a certain task, such as setting up the instrument to take a measurement.

Table 2-5: Alias Commands

Header	Description
ALias?	Query the alias state
ALias:CATALOG?	Return a list of aliases
ALias:DEFINE	Query/set (create) a new alias
ALias:DELEte	Remove an alias
ALias:DELEte:ALL	Remove all aliases
ALias:DELEte:NAME	Remove a named alias
ALias:STATE	Query/set the alias state on and off

Calibration and Diagnostic Commands

Calibration and Diagnostic commands let you initiate the self-calibration and diagnostic routines that are built into the instrument. The diagnostic test operation includes selecting the test sequence, executing the sequence, and then examining the results.

Table 2-6: Calibration and Diagnostic Commands

Header	Description
CALibrate:CONTINUE	Execute next factory calibration step
CALibrate:FACTory	Initialize factory calibration sequence
CALibrate:INTERNAL	Perform an internal signal path compensation
CALibrate:STATUS?	Return status from last calibration sequence

Table 2–6: Calibration and Diagnostic Commands (Cont.)

Header	Description
DIAG:RESULT:FLAG?	Return diagnostic tests status
DIAG:RESULT:LOG?	Return diagnostic test sequence results
DIAG:SELEct:ALL	Select complete diagnostic test sequence for acquisition, CPU, display, and front panel
DIAG:STATE:EXECute	Control diagnostic tests

Cursor Commands

Cursor commands provide control over the oscilloscope cursor display and readout.

Table 2–7: Cursor Commands

Header	Description
CURSor?	Returns cursor settings
CURSor:FUNction	Query/set cursors on or off; select cursor type
CURSor:HBArs?	Return horizontal bar settings
CURSor:HBArs:DELTA?	Return vertical distance between horizontal bar cursors
CURSor:HBArs:POSITION<x>	Query/set position of a horizontal bar cursor
CURSor:HBArs:SELEct	Query/set which horizontal bar cursor is active
CURSor:PAIred?	Return paired cursor position measurements
CURSor:PAIred:HDELTA?	Query horizontal distance between first and second paired cursors

Table 2–7: Cursor Commands (Cont.)

Header	Description
CURSor:PAIred:HPOS1?	Query horizontal position of first paired cursor
CURSor:PAIred:HPOS2?	Query horizontal position of second paired cursor
CURSor:PAIred:POSITION<x>	Query/set vertical bar position of paired cursor
CURSor:PAIred:SElect	Query/set active paired cursor
CURSor:PAIred:VDELTA?	Query vertical distance between first and second paired cursors
CURSor:VBArS?	Return vertical bar settings
CURSor:VBArS:DELTA?	Return horizontal distance between cursors
CURSor:VBArS:POSITION<x>	Query/set position of a vertical bar cursor
CURSor:VBArS:SElect	Query/set which vertical bar cursor is active
CURSor:VBArS:UNIts	Query/set vertical cursors to time or frequency

Display Commands

Display commands let you change the graticule style, change the displayed contrast, and clear the menu.

Table 2–8: Display Commands

Header	Description
CLEARMenu	Clear menus from display
DISplay?	Returns display settings
DISplay:CONTRast	Query/set the LCD display contrast

Table 2–8: Display Commands (Cont.)

Header	Description
DISplay:FORMat	Query/set YT or XY display
DISplay:GRAticule	Query/set graticule style
DISplay:MODe	Query/set the functional mode of the instrument: scope or meter mode
DISplay:PERSiStence	Query/set the accumulate time
DISplay:STYle	Query/set dots, vectors, dot accumulate or vector accumulate
DISplay:TRIGT	Query/set the display of the trigger indicator on screen

Hard Copy Commands

The hard copy commands let you control the format of hard copy output and control the initiation and termination of hard copies.

Table 2–9: Hard Copy Commands

Header	Description
HARDCopy	Start or terminate hard copy
HARDCopy:CLEARSpool	Empty hard copy spooler
HARDCopy:FORMat	Query/set the hard copy output format
HARDCopy:LAYout	Query/set the hard copy orientation
HARDCopy:PORT	Query/set the hard copy port for output (only RS232 allowed)

Horizontal Commands

Horizontal commands control the time bases of the instrument. You can set the time per division (or time per point) of both the main and delayed time bases.

Table 2–10: Horizontal Commands

Header	Description
HORizontal?	Return horizontal settings
HORizontal:DELay?	Return delay time base settings
HORizontal:DELay:MODE	Query/set delay time base mode
HORizontal:DELay:SCALE	Query/set delay time base time/division
HORizontal:DELay:SECdiv	Same as HORizontal:DELay:SCALE
HORizontal:DELay:TIME	Query/set delay time
HORizontal:DELay:TIME?	Return delay time parameters
HORizontal:DELay:TIME:RUNS-After	Query/set time to wait in delay-runs-after-main mode
HORizontal:FITtoscreen	Query/set horizontal 10X magnify
HORizontal:MAIn	Query/set main time base time/division
HORizontal:MAIn:SCALE	Query/set main time base time/division
HORizontal:MAIn:SECdiv	Same as HORizontal:MAIn:SCALE
HORizontal:MODE	Query/set delay time base on or off
HORizontal:POSition	Query/set position of waveform to display
HORizontal:RECOrdlength?	Return number of points in waveform record (always 2500 points)
HORizontal:REF<x>	Query/set position lock for REF waveforms
HORizontal:SCALE	Same as HORizontal:MAIn:SCALE
HORizontal:SECdiv	Same as HORizontal:MAIn:SCALE

Table 2–10: Horizontal Commands (Cont.)

Header	Description
HORizontal:TRIGger?	Query main time base trigger position
HORizontal:TRIGger:POSition	Query/set main time base trigger position

Oscilloscope Measurement Commands

Measurement commands control the automated measurement system. Up to four automated measurements can be displayed on the instrument screen. In the commands, these four measurement readouts are named MEAS<x>, where <x> can be 1, 2, 3, or 4.

In addition to the four measurement readouts displayed, the measurement commands let you specify a fifth measurement, IMMEd. The immediate measurement has no front-panel equivalent, and the instrument never displays immediate measurements. Because they are computed only when they are requested, immediate measurements slow the waveform update rate less than displayed measurements.

Use the VALue? query to obtain measurement results of either displayed or immediate measurements.

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

Table 2–11: Oscilloscope Measurement Commands

Header	Description
MEASUrement?	Return all measurement parameters
MEASUrement:GATing	Query/set measurement gating
MEASUrement:IMMed?	Return immediate measurement parameters

Table 2–11: Oscilloscope Measurement Commands (Cont.)

Header	Description
MEASUrement:IMMed:SOUrce	Query/set channel to take the immediate measurement from
MEASUrement:IMMed:TYPe	Query/set the immediate measurement to be taken
MEASUrement:IMMed:UNIts?	Return the immediate measurement units
MEASUrement:IMMed:VALue?	Return the immediate measurement result
MEASUrement:MEAS<x>?	Return parameters on the measurement
MEASUrement:MEAS<x>:SOUrce	Query/set channel to take the measurement from
MEASUrement:MEAS<x>:STATE	Query/set measurement display on or off
MEASUrement:MEAS<x>:TYPe	Query/set the type of measurement to be taken
MEASUrement:MEAS<x>:UNIts?	Returns the units to use for measurement
MEASUrement:MEAS<x>:VALue?	Returns measurement results
MEASUrement:METhod	Query/set method for calculating reference levels
MEASUrement:REFLevel?	Returns percent and absolute reference levels
MEASUrement:REFLevel:ABSolute:HIGH	Query/set the top level for measurement (90% level)
MEASUrement:REFLevel:ABSolute:LOW	Query/set the low level for measurement (10% level)
MEASUrement:REFLevel:ABSolute:MID	Query/set the mid level for measurements

Table 2–11: Oscilloscope Measurement Commands (Cont.)

Header	Description
MEASUREMENT:REFLEVEL:METHOD	Query/set the method to assign HIGH, MID and LOW levels: either % or absolute volts
MEASUREMENT:REFLEVEL:PERCENT:HIGH	Query/set the top level for measurement (90% level)
MEASUREMENT:REFLEVEL:PERCENT:LOW	Query/set the low level for measurement (10% level)
MEASUREMENT:REFLEVEL:PERCENT:MID	Query/set the mid level for measurements

Miscellaneous Oscilloscope Commands

Miscellaneous oscilloscope commands are a group of commands that do not fit into any other category.

Several commands and queries used with the instrument are common to all devices on the RS-232 interface. These commands and queries are defined by IEEE Std. 488.2-1987 and Tek Standard Codes and Formats 1989 and begin with an asterisk (*) character.

Table 2–12: Miscellaneous Oscilloscope Commands

Header	Description
AUTOSet	Automatic instrument setup
AUTORange	Query/set autorange mode on or off
*DDT	Query/set group execute trigger (GET)
FACTory	Reset to factory default
HDR	Same as HEADER
HEADER	Query/set command header
ID?	Return Tektronix Codes and Formats instrument identification

Table 2–12: Miscellaneous Oscilloscope Commands (Cont.)

Header	Description
*IDN?	Return IEEE 488.2 instrument identification
*LRN?	Learn device setting
LOCK	Query/set front panel lock
NEWpass	Change password for User Protected Data
PASSWord	Access to change User Protected Data
POWER:BACKlight	Query/set inactivity timer for backlight
POWER:STANDBY	Query/set inactivity timer for automatic power off.
*PUD	Query/set user-protected data
REM	No action; remark only
*RST	Reset most instrument settings to factory default values
SET?	Same as *LRN?
TEKSecure	Initialize waveforms and setups
*TRG	Perform Group Execute Trigger (GET)
*TST?	Return self-test results
UNLOCK	Unlock front panel (local lockout)
VERBose	Query/set full command name or minimum spellings for query responses

RS-232 Commands

RS-232 commands allow you to set or query the parameters that control the RS-232 port.

Table 2–13: RS-232 Commands

RS232?	Query RS232 parameters
RS232:BAUd	Query/set baud rate
RS232:CONTR01?	Returns flow control parameters
RS232:CONTR01:RTS	Query/set hard flagging
RS232:HARDFlagging	Query/set hard flagging
RS232:MODE	Query/set data transmission mode, query always returns RAW
RS232:PACE	Query/set soft flagging
RS232:PARity	Query/set parity type
RS232:PRESet	Set default RS-232 parameters
RS232:SBITS	Query/set number of stop bits
RS232:SOFTFlagging	Query/set soft flagging
RS232:STOPBits	Query/set number of stop bits
RS232:TRANsmit:DELay	Query/set delay before query response
RS232:TRANsmit:TERMinator	Query/set end-of-line terminator

Oscilloscope Save and Recall Commands

Save and Recall commands allow you to store and retrieve internal waveforms and settings. When you “save a setting,” you save most of the settings of the instrument. When you then “recall a setting,” the instrument restores itself to the state it was in when you saved that setting.

Table 2–14: Oscilloscope Save and Recall Commands

Header	Description
*RCL	Recall setting
RECALL:SETup	Recall saved instrument setting
REFA:WAVEform	Recall stored waveform into Ref A
REFB:WAVEform	Recall stored waveform into Ref B
*SAV	Save instrument setting
SAVE:SETup	Save instrument setting
SAVE:WAVEform	Save waveform

Status and Error Commands

Status and error commands let you determine the status of the instrument and control events.

Several commands and queries used with the instrument are common to all devices on the RS-232 interface. These commands and queries are defined by IEEE Std. 488.2-1987 and Tek Standard Codes and Formats 1989, and begin with an asterisk (*) character.

Table 2–15: Status and Error Commands

Header	Description
ALLEv?	Return all events
BUSY?	Return instrument busy status
*CLS	Clear status
DESE	Query/set device event status enable
*ESE	Query/set standard event status enable
*ESR?	Return standard event status register
EVENT?	Return event code

Table 2–15: Status and Error Commands (Cont.)

Header	Description
EVMsg?	Return event message
EVQty?	Return number of events in queue
*OPC	Query/set operation complete
*PSC	Query/set power-on status clear
*SRE	Query/set service request enable
*STB?	Read status byte
*WAI	Wait to continue

Trigger Commands

Trigger commands control all aspects of instrument triggering.

You can set the main trigger to one of three modes: edge, pulse, or video. Edge triggering is the default mode.

Edge triggering lets you acquire a waveform when the signal passes through a voltage level of your choosing. Pulse triggering allows you to trigger on glitches or specific pulse widths. Video triggering adds the capability of triggering on video fields and lines.

Table 2–16: Trigger Commands

Header	Description
TRIGger?	Return trigger settings
TRIGger	Force trigger event
TRIGger:MAIn	Set main trigger level to 50%; Query returns main trigger level
TRIGger:MAIn:EDGE?	Return edge trigger parameters
TRIGger:MAIn:EDGE:COUPling	Query/set edge trigger coupling
TRIGger:MAIn:EDGE:SLOpe	Query/set edge trigger slope

Table 2–16: Trigger Commands (Cont.)

Header	Description
TRIGger:MAIn:EDGE:SOUrce	Query/set edge trigger source
TRIGger:MAIn:HOLDOff?	Return trigger holdoff value
TRIGger:MAIn:HOLDOff:VALue	Query/set trigger holdoff value
TRIGger:MAIn:LEVe1	Query/set trigger level
TRIGger:MAIn:MODE	Query/set trigger mode
TRIGger:MAIn:PULse:EQUAL	Query/set pulse width equality tolerance
TRIGger:MAIn:PULse:HOLDOff?	Return pulse trigger holdoff value
TRIGger:MAIn:PULse:HOLD-Off:VALue	Query/set pulse trigger holdoff value
TRIGger:MAIn:PULse:NOTEqual	Query/set pulse width inequality tolerance
TRIGger:MAIn:PULse:POLarity	Query/set pulse width polarity
TRIGger:MAIn:PULse:SOUrce	Query/set pulse trigger source
TRIGger:MAIn:PULse:WHEn	Query/set pulse trigger condition
TRIGger:MAIn:PULse:WIDth	Query/set nominal pulse width for pulse trigger
TRIGger:MAIn:TYPe	Query/set main trigger type
TRIGger:MAIn:VIDeo:FIELD	Query/set video trigger field
TRIGger:MAIn:VIDeo:HOLDOff?	Return video trigger holdoff
TRIGger:MAIn:VIDeo:HOLD-Off:VALue	Query/set video trigger holdoff value
TRIGger:MAIn:VIDeo:SCAN	Query/set video trigger scan rate
TRIGger:MAIn:VIDeo:SOUrce	Query/set video trigger source
TRIGger:STATE?	Return trigger system status

Vertical Commands

Vertical commands control the display of channels and of math and reference waveforms. The `SElect:<wfm>` command also selects the waveform to be used by many commands in other command groups.

Table 2–17: Vertical Commands

Header	Description
<code>CH<x>?</code>	Return vertical parameters
<code>CH<x>:BANdwidth</code>	Query/set channel bandwidth
<code>CH<x>:COUPling</code>	Query/set channel coupling
<code>CH<x>:INVert</code>	Query/set invert channel
<code>CH<x>:POSition</code>	Query/set channel position
<code>CH<x>:PRObe?</code>	Return channel probe parameters
<code>CH<x>:PROBe:CURrentscale</code>	Query/set channel current probe conversion factor
<code>CH<x>:PROBe:UNITs</code>	Query/set channel probe units
<code>CH<x>:PROBe:VOLTSCALE</code>	Query/set channel voltage probe attenuation
<code>CH<x>:SCALE</code>	Query/set channel volts/div
<code>CH<x>:VOLts</code>	Same as <code>CH<x>:SCALE</code>
<code>MATH?</code>	Same as <code>MATH1:DEFINE</code>
<code>MATH1:DEFINE</code>	Query/set math waveform definition
<code>SElect?</code>	Return selected waveform
<code>SElect:<wfm></code>	Query/set selected waveform
<code>SElect:CONTROL</code>	Query/set the channel affected by the front-panel controls

Table 2–17: Vertical Commands (Cont.)

Header	Description
Z00m:VERTical:POSition	Query/set the vertical position of math and reference waveforms
Z00m:VERTical:SCAlE	Query/set the vertical expansion and compression factor of math and reference waveforms

Waveform Commands

Waveform commands let you transfer waveform data points to and from the instrument. Waveform data points are a collection of values that define a waveform. One data value usually represents one data point in the waveform record. When working with enveloped waveforms, each data value is either the min or max of a max/min pair. Before you can transfer waveform data, you must specify the data format, record length, and waveform locations.

Waveform Data Formats. DMM logger data always uses one 8-bit data byte to represent each point. Oscilloscope waveform data uses either one or two 8-bit data bytes to represent each data point. The number of bytes used depends on the acquisition mode specified when you acquired the data. Data acquired in SAMple, ENVELOpe, or PEAKdetect modes uses one 8-bit byte per waveform data point; data acquired in AVERage mode uses two 8-bit bytes per point. For more information on the acquisition modes see the ACQUIRE:MODE command on page 2–40.

The DATA:WIDTH command lets you specify the number of bytes per data point when transferring data to and from the instrument. If you specify two bytes for data that uses only one, the least significant byte will be filled with zeros; if you specify one byte for data that uses two, the least significant byte will be ignored.

The instrument can transfer waveform data in either ASCII or binary format. Use the `DATA:ENCdg` command to specify one of the following formats:

- ASCII data is represented by single byte, signed integer values in the range -128 to 127 . Each data point value consists of up to five ASCII characters for the value and one for the minus sign if the value is negative. Commas separate data points.

An example of an ASCII waveform data string follows:

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

- Binary data can be 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 .

The defined binary formats also specify the order in which the bytes are transferred giving a total of four binary formats: 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/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. The `DATA:START` and `DATA:STOP` commands let you specify the first and last data points of the waveform record.

When transferring data into the instrument you must specify the location of the first data point within the waveform record. For example, when `DATA:START` is set to 1, data points will be stored starting with the first point in the record, and when `DATA:START` is set to 500, data will be stored starting at the 500th point in the record.

DATA:STOP will be ignored when transferring data into the instrument as the instrument will stop reading data when there is no more data to read or when the record length has been reached.

When transferring data from the instrument you must specify the first and last data points in the waveform record. Setting DATA:START to 1 and DATA:STOP to 2500 always returns the entire waveform.

Waveform Data Locations and Memory Allocation. The DATA:SOURce command specifies the location of the data when transferring waveforms from the instrument. You can transfer multiple oscilloscope waveforms at one time by specifying more than one source. You can only transfer the displayed DMM logger waveform.

You can transfer only one oscilloscope waveform to the instrument at a time. Each oscilloscope waveform sent to the instrument is stored in one of ten stored waveform locations. You specify the stored waveform location with the DATA:DESTination command. The waveform should be 2500 data points in length.

NOTE. *The instrument accepts waveforms that are ≤ 2500 data points long. The instrument will truncate waveforms longer than 2500 data points.*

Waveform Preamble. Each waveform that is transferred has an associated waveform preamble that contains information such as the horizontal scale, vertical scale, and other settings in place when the waveform was created. Refer to the WFMPre commands starting on page 2–239 for more information about the waveform preamble.

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. The GETWFM program on the diskette supplied with this manual shows you how to scale data.

Transferring Waveform Data From the Instrument. Transfer waveforms from the instrument to an external controller using the following sequence.

1. Select the waveform source(s) using the `DATA:SOURce` command. If you want to transfer multiple waveforms, select more than one source.
2. Specify the waveform data format using `DATA:ENCdg`.
3. Specify the number of bytes per data point using `DATA:WIDth`.
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 `WFMPRe?` or `DMM:WFMPRe?` query.
6. Transfer waveform data from the instrument using the `CURVe?` or `DMM:CURVE?` query.

Transferring Waveform Data to the Instrument. Transfer waveform data to one of ten stored waveform locations in the instrument using the following sequence.

1. Specify the stored waveform location using `DATA:DESTination`.
2. Specify the waveform data format using `DATA:ENCdg`.
3. Specify the number of bytes per data point using `DATA:WIDth`.
4. Specify the first data point in the waveform record using `DATA:STARt`.
5. Transfer waveform preamble information using `WFMPRe:<wfm>`.
6. Transfer waveform data to the instrument using `CURVe`.

Table 2–18: Waveform Commands

Header	Description
CURVe	Transfer waveform data from the instrument
DATA:DESTination	Query/set the destination for waveforms sent to the instrument
DATA:ENCdg	Query/set the waveform data encoding method
DATA:SOURce	Query/set the source of CURVe? data
DATA:START	Query/set the starting point in waveform transfer
DATA:STOP	Query/set the ending point in waveform transfer
DATA:TARget	Same as DATA:DESTination
DATA:WIDth	Query/set the byte width of waveform points
WAVFrm?	Returns waveform preamble and curve data
WFMPre?	Returns waveform preamble
WFMPre:BIT_Nr	Query/set the preamble bit width of waveform points
WFMPre:BN_Fmt	Query/set the preamble binary encoding type
WFMPre:BYT_Nr	Query/set the preamble byte width of waveform points
WFMPre:BYT_Or	Query/set the preamble byte order of waveform points
WFMPre:ENCdg	Query/set the preamble encoding method
WFMPre:PT_Fmt	Set the format of curve points (always ENVELOPE for DMM)
WFMPre:PT_Off	Set the trigger position

Table 2–18: Waveform Commands (Cont.)

Header	Description
WFMPre:XINcr	Set the horizontal sampling interval
WFMPre:YMUlt	Set the vertical scale factor
WFMPre:YOff	Set the vertical offset
WFMPre:YZero	Set the offset voltage
WAVPre:<wfm>?	Returns waveform formatting data
WFMPre:<wfm>:NR_Pt	Query/set the number of points in the curve
WFMPre:<wfm>:PT_Fmt	Query/set the format of curve points
WFMPre:<wfm>:PT_Off	Query/set the trigger position
WFMPre:<wfm>:WFId?	Query the curve identifier
WFMPre:<wfm>:XINcr	Query/set the horizontal sampling interval
WFMPre:<wfm>:XUNit?	Query the horizontal units
WFMPre:<wfm>:YMUlt	Query/set the vertical scale factor
WFMPre:<wfm>:YOff	Query/set the vertical position
WFMPre:<wfm>:YUNit?	Query the vertical units
WFMPre:<wfm>:YZero	Query/set the offset voltage

DMM Commands

The following command groups cover functions specific to meter-mode operation.

Miscellaneous DMM Commands

Miscellaneous DMM commands are a group of commands that do not fit into any other category.

Table 2–19: Miscellaneous DMM Commands

Header	Description
DMM:DISPlay:DELTA	Query/set the readouts/logger with respect to Rel Δ reference
DMM:DISPlay:GRAticule	Query/set the graticule style
DMM:DISPlay:STYle	Query/set the vertical thickness of the data logger plot
DMM:FREQuency:REJect	Query/set the DMM frequency reject value
DMM:FUNCTion	Query/set the DMM function
DMM:HOLD	Query/set the DMM hold function
DMM:RESet	Reset DMM logger and measurements
DMM:STATE?	Return run/hold/auto status

DMM Calibration Commands

These commands control external calibration of the DMM.

Table 2–20: DMM Calibration Commands

Header	Description
DMM:CALibrate:CONTINUE	Move to next calibration step
DMM:CALibrate:FACTory	Initialize factory calibration
DMM:CALibrate:STATUS?	Return pass/fail status of last calibration step

DMM Function Range Commands

These commands set and query the range for each of the DMM functions.

Table 2–21: DMM Function Range Commands

Header	Description
DMM:RANGE:ACV	Query/set the AC voltage range
DMM:RANGE:AUTO	Query/set autoranging on and off
DMM RANGE:DCV	Query/set the DC voltage range
DMM:RANGE:OHMS	Query/set the resistance range

DMM Measurement Commands

Measurement commands control the DMM statistic system. Up to three statistics can be displayed on the screen of the instrument.

Table 2–22: DMM Measurement Commands

Header	Description
DMM:DELTA:VALue	Query/set the Rel Δ reference value
DMM:MEASUrement?	Return all DMM measurement parameters
DMM:MEASUrement:AVErage?	Return DMM Avg statistic

Table 2–22: DMM Measurement Commands (Cont.)

Header	Description
DMM:MEASUrement:CONTInuity?	Return continuity measurement
DMM:MEASUrement:DELTA?	Return DMM Rel Δ statistic
DMM:MEASUrement:MAX?	Return DMM Max statistic
DMM:MEASUre- ment:MEAS<x>:STATE	Query/set specific DMM measure- ments to on or off
DMM:MEASUre- ment:MEAS<x>:TYPE	Query/set a specific DMM statistic type
DMM:MEASUre- ment:MEAS<x>:VALue?	Return the value of the specified statistic
DMM:MEASUrement:MIN?	Return DMM Min statistic
DMM:MEASUrement:PK2pk?	Return DMM Max–Min statistic
DMM:VALue?	Return the current DMM reading

DMM Probe Commands

These commands set and query the probe type and scale factor.

Table 2–23: DMM Probe Commands

Header	Description
DMM:PROBe:CURrentscalee	Query/set the DMM current probe conversion factor
DMM:PROBe:UNITs	Query/set the DMM probe unit

DMM Beeper Commands

These commands set and query the function of the DMM beeper.

Table 2–24: DMM Beeper Commands

Header	Description
DMM:BEEP:CONTInuity	Query/set the beeper for continuity function
DMM:BEEP:NEWMax	Query/set the beeper for new Max and Min statistics

DMM Logger Commands

These commands set and query the data logger and its cursor functions.

Table 2–25: DMM Logger Commands

Header	Description
DMM:CURSor:FUNcTion	Query/set the DMM cursor function
DMM:CURSor:HBARs?	Returns the DMM horizontal bar cursor positions
DMM:CURSor:HBARs:DELtA?	Returns the DMM horizontal bar cursor delta
DMM:CURSor:HBARs:POSI-TION<x>	Query/set the DMM horizontal bar cursor position
DMM:CURSor:HBARs:SElect	Query/set the active DMM cursor
DMM:CURSor:PAIred?	Returns the DMM paired cursor positions
DMM:CURSor:PAIred:HDELtA?	Returns the vertical distance between DMM paired cursors
DMM:CURSor:PAIred:HPOS1?	Returns the vertical position of the first DMM paired cursor
DMM:CURSor:PAIred:HPOS2?	Returns the vertical position of the second DMM paired cursor
DMM:CURSor:PAIred:POSI-TION<x>	Query/set the DMM paired cursor positions
DMM:CURSor:PAIred:SElect	Query/set the active DMM paired cursor
DMM:CURSor:PAIred:VDELtA?	Returns the time distance between the DMM paired cursors
DMM:CURSor:VBARs?	Set/query the DMM vertical bar cursor positions
DMM:CURSor:VBARs:DELtA?	Return the time between the DMM vertical bar cursors

Table 2–25: DMM Logger Commands (Cont.)

Header	Description
DMM:CURSor:VBARs:POSITION<x>	Query/set the DMM vertical bar cursor positions
DMM:CURSor:VBARs:SElect	Query/set the active DMM vertical bar cursor
DMM:CURVe?	Transfer data logger data
DMM:LOGGer:MODe	Query/set the DMM logger acquisition mode
DMM:LOGGer:SECDiv	Query/set DMM logger horizontal scale
DMM:WAVFrm?	Combines the DMM:WFMPre? and DMM:CURVe? queries
DMM:WFMPre?	Returns the data logger preamble
DMM:WFMPRe:WFId?	Returns information about the data logger waveform

DMM Save and Recall Commands

These commands set and query the range for each of the DMM functions.

Table 2–26: DMM Save and Recall Commands

Header	Description
DMM:RECA11:DATA	Recalls stored DMM data to the display
DMM:SAVe:DATA	Saves DMM data

Command Descriptions

Commands either set oscilloscope features or query oscilloscope values. You can use some commands to do both, some to only set, and some to only query. This manual marks set only commands with the words “No Query Form” included with the command name. It marks query only commands with a question mark appended to the header and includes the words “Query Only” in the command name.

This manual fully spells out headers, mnemonics, and arguments with the minimal spelling shown in upper case. For example, to use the abbreviated form of the ACQUIRE:MODE command just type ACQ:MOD.

ACQUIRE? (Query Only)

Returns all the current acquisition parameters.

Group

Acquisition

Syntax

ACQUIRE?

Returns

None

Examples

ACQUIRE?

might return the string :ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16 for the current acquisition parameters.

Related Commands

None

ACQUIRE:MODE

Sets or queries the oscilloscope acquisition mode of the instrument. This affects all live waveforms. This command is equivalent to setting Mode in the Acquire menu.

Waveforms are the displayed data point values taken from acquisition intervals. Each acquisition interval represents a time duration that is determined by the horizontal scale (time per division). The instrument sampling system always samples at the maximum rate, and so an acquisition interval may include more 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.

Group

Acquisition

Syntax

```
ACQUIRE:MODE { SAMPLE | PEAKdetect | AVERAGE |  
ENVELOPE }
```

```
ACQUIRE:MODE?
```

Arguments

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

PEAKdetect specifies the display of the high-low range of the samples taken from a single waveform acquisition. The oscilloscope displays the high-low range 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.

AVERAGE specifies averaging mode, where the resulting waveform shows an average of SAMPLE data points from several separate waveform acquisitions. The number of waveform acquisitions that go

into making up the average waveform is set or queried using the ACQUIRE:NUMAVg command.

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

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 ENVELOPE.

Related Commands

ACQUIRE:NUMAVg, ACQUIRE:NUMENV, CURVe?, DATA:WIDTH

ACQUIRE:NUMACq? (Query Only)

Indicates the number of acquisitions that have taken place since starting oscilloscope acquisition. This value is reset to zero when any Acquisition, Horizontal, or Vertical arguments that affect the waveform are modified. The maximum number of acquisitions that can be counted is $2^{30}-1$. This value is the same value that the oscilloscope displays at the top of the screen.

Group

Acquisition

Syntax

```
ACQUIRE:NUMACq?
```

Returns

<NR1>

Examples

ACQUIRE:NUMACQ?

might return 350, indicating that 350 acquisitions took place since an ACQUIRE:STATE RUN command was executed.

Related Commands

ACQUIRE:STATE

ACQUIRE:NUMAVG

Sets the number of oscilloscope waveform acquisitions that make up an averaged waveform. This command is equivalent to setting the Average count in the Acquire Mode pop-up menu.

Group

Acquisition

Syntax

ACQUIRE:NUMAVG <NR1>

ACQUIRE:NUMAVG?

Arguments

<NR1> is the number of waveform acquisitions, from 2 to 256.

Examples

ACQUIRE:NUMAVG 10

specifies that an averaged waveform will show the result of combining 10 separately acquired waveforms.

ACQUIRE:NUMAVG?

might return 75, indicating that there are 75 acquisitions specified for averaging.

Related Commands

ACQUIRE:MODE

ACQUIRE:NUMENV

Sets the number of oscilloscope waveform acquisitions that make up an envelope waveform. This command is equivalent to setting the Envelope count in the Acquire Mode pop-up menu.

Group

Acquisition

Syntax

```
ACQUIRE:NUMENV { <NR1> | INFINITE }
```

```
ACQUIRE:NUMENV?
```

Arguments

<NR1> \neq 0 is the number of waveform acquisitions, from 2 to 256. The envelope will restart after the specified number of envelopes have been acquired or when the ACQUIRE:STATE RUN command is sent.

INFINITE or <NR1> = 0 specifies continuous enveloping.

NOTE. *If you set the acquisition system to single sequence envelope mode and set the number of envelopes to infinity, the instrument will envelope a maximum of 256 acquisitions.*

Examples

```
ACQUIRE:NUMENV 10
```

specifies that an enveloped waveform will show the result of combining 10 separately acquired waveforms.

```
ACQUIRE:NUMENV?
```

might return 0, indicating that acquisitions are acquired infinitely for enveloped waveforms.

Related Commands

```
ACQUIRE:MODE
```

ACQUIRE:STATE

Starts or stops oscilloscope acquisitions. This command is the equivalent of pressing the front-panel HOLD button. If ACQUIRE:STOPAfter is set to SEQUENCE, other signal events may also stop acquisition.

Group

Acquisition

Syntax

```
ACQUIRE:STATE { OFF | ON | RUN | STOP | <NR1> }
```

```
ACQUIRE:STATE?
```

Arguments

OFF or STOP or <NR1> = 0 stops acquisitions.

ON or RUN or <NR1> \neq 0 starts acquisition and display of waveforms. If the command was issued in the middle of an acquisition sequence (for instance averaging or enveloping), RUN restarts the sequence, discarding any data accumulated before the STOP. It also resets the number of acquisitions.

Examples

```
ACQUIRE:STATE RUN
```

starts acquisition of waveform data and resets the number of acquisitions count (NUMACq) to zero.

```
ACQUIRE:STATE?
```

returns either 0 or 1, depending on whether the acquisition system is running.

Related Commands

ACQUIRE:NUMACq?, ACQUIRE:STOPAfter

ACQUIRE:STOPAFTER

Tells the instrument when to stop taking oscilloscope acquisitions. This command is equivalent to setting Stop After in the Acquire menu.

Group

Acquisition

Syntax

```
ACQUIRE:STOPAFTER { RUNSTOP | SEQUENCE }
```

```
ACQUIRE:STOPAFTER?
```

Arguments

RUNSTOP specifies that the run and stop state should be determined by the user pressing the front-panel HOLD button.

SEQUENCE specifies “single sequence” operation, where the instrument stops after it has acquired enough waveforms to satisfy the conditions of the acquisition mode. For example, if the acquisition mode is set to sample, the instrument stops after digitizing a waveform from a single trigger event. However, if the acquisition mode is set to average 16 waveforms, then the instrument stops only after acquiring all 16 waveforms. The ACQUIRE: STATE command and the front-panel RUN/STOP button also stop acquisitions when the instrument is in single sequence mode.

NOTE. *If you set the acquisition system to single sequence, envelope mode, and set the number of envelopes to infinity, the instrument will envelope a maximum of 257 acquisitions.*

Examples

```
ACQUIRE:STOPAFTER RUNSTOP
```

sets the oscilloscope to stop acquisition when the user presses the front-panel HOLD button.

```
ACQUIRE:STOPAFTER?
```

might return SEQUENCE.

Related Commands

ACQuire:MODE, ACQuire:STATE

ALias

Turns command aliases on or off. This command is identical to the ALias:STATE command.

Group

Alias

Syntax

ALias { OFF | ON | <NR1> }

ALias?

Arguments

OFF or <NR1> = 0 turns alias expansion off. If a defined alias label is sent when ALias is OFF, an execution error will be generated.

ON or <NR1> \neq 0 turns alias expansion on. When the oscilloscope receives a defined alias, it substitutes the specified command sequence for the alias and executes it.

Examples

ALIAS ON

turns the alias feature on.

ALIAS?

returns 1 when aliases are on.

Related Commands

None

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?

Returns

<QString>[,<QString>...]

Examples

ALIAS:CATALOG?

might return the string "SETUP1", "TESTMENU1", "DEFAULT", showing there are 3 aliases named SETUP1, TESTMENU1, and DEFAULT.

Related Commands

None

ALIAS:DEFINE

Assigns a sequence of program messages to an alias label. If ALIAS:STATE has been turned ON, these messages are substituted for the alias whenever it is received as a command or query. The ALIAS:DEFINE? query returns the definition of a selected alias.

Group

Alias

Syntax

ALIAS:DEFINE <QString><Comma>{ <QString> | <Block> }

ALIAS:DEFINE? <QString>

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 ≤12 characters.

The second <QString> or <Block> is a complete sequence of program messages. The messages can contain only valid commands separated by semicolons and following all rules for concatenating commands (see page 2–4). The sequence must be ≤80 characters.

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

Examples

```
ALIAS:DEFINE "ST1",":RECALL:SETUP 5;:AUTOSET  
EXECUTE;:SELECT:CH1 ON"
```

defines an alias named “ST1” that sets up the instrument.

```
ALIAS:DEFINE? "ST1"  
might return :ALIAS:DEFINE "ST1",#239:RECALL:SETUP  
5;:AUTOSET EXECUTE;:SELECT:CH1 ON
```

Related Commands

None

ALIAS:DELETE (No Query Form)

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

Group

Alias

Syntax

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

Arguments

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

Examples

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

Related Commands

None

ALIAS:DELETE:ALL (No Query Form)

This command deletes all existing aliases.

Group

Alias

Syntax

```
ALIAS:DELETE:ALL
```

Arguments

None

Examples

```
ALIAS:DELETE:ALL
deletes all aliases.
```

Related Commands

None

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 a previously defined alias.

Examples

ALIAS:DELETE:NAME "STARTUP"
deletes the alias named **STARTUP**.

Related Commands

None

ALIAS:STATE

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

Group

Alias

Syntax

ALIAS:STATE { OFF | ON | <NR1> }

ALIAS:STATE?

Arguments

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

ON or <NR1> \neq 0 turns alias expansion on. When the oscilloscope receives a defined alias, it substitutes the specified command sequence for the alias and executes it.

Examples

ALIAS:STATE OFF
turns the command alias feature off.

ALIAS:STATE?
returns 0 when alias mode is off.

Related Commands

None

ALLEv? (Query Only)

Causes the instrument to return all events and their messages, and removes the returned events from the Event Queue. The messages are separated by commas. Use the *ESR? query to enable the events to be returned. For a complete discussion of the use of these registers, see page 3–1. This command is similar to repeatedly sending *EVMsg? queries to the instrument.

Group

Status and error

Syntax

ALLEv?

Returns

The event code and message in the following format:

<Event Code><Comma><QString> [<Comma><Event Code><Comma><QString>...]

<QString> ::= <Message>; [<Command>]

<Command> is the command that caused the error and may be returned when a command error is detected by the instrument. As much of the command is returned as possible without exceeding the 60 character limit of the <Message> and <Command> strings combined. The command string is right-justified.

Examples

ALLEV?

might return the string :ALLEV 2225,"Measurement error, No waveform to measure; ",420,"Query UNTERMINATED; ".

Related Commands

*CLS, DESE, *ESE, *ESR?, EVENT?, EVMsg?, EVQty?, *SRE, *STB?

AUTORange (No Query Form)

Causes the instrument to adjust its scope-mode vertical, horizontal, and trigger controls to provide a stable display of the displayed waveforms. This command is equivalent to pressing the front-panel AUTORANGE button while in scope mode. For a detailed description of the autorange function, consult the *THS 710 & THS 720 User Manual*.

Group

Miscellaneous Oscilloscope

Syntax

AUTORange { OFF | ON | <NR1> }

Arguments

OFF or <NR1> = 0 turns off the autorange function.

ON or <NR1> ≠ 0 turns on the autorange function.

Examples

None

Related Commands

AUTOSet

AUTOSet (No Query Form)

Causes the instrument to adjust its scope-mode vertical, horizontal, and trigger controls to provide a stable display of the displayed live waveforms. Autoset does not effect displayed reference waveforms. This function is a one-time autoset and does not have a front-panel equivalent.

NOTE. *The AUTOSet command does not return control to the instrument controller until the autoset operation is complete.*

Group

Miscellaneous Oscilloscope

Syntax

AUTOSet { EXECute }

Arguments

EXECute autosets the displayed waveforms one time.

Examples

None

Related Commands

AUTORange

BUSY? (Query Only)

Returns the status of the instrument. This command allows you to synchronize the operation of the instrument with your application program. Synchronization methods are described on page 3–9.

Group

Status and error

Syntax

BUSY?

Returns

<NR1> = 0 means that the instrument is not busy processing a command whose execution time is extensive. These commands are listed in Table 2–27.

<NR1> = 1 means that the instrument is busy processing one of the commands listed in Table 2–27.

Table 2–27: Commands that Affect BUSY? Response

Operation	Command
Single sequence acquisition	ACQuire:STATE ON or ACQuire:STATE RUN (when ACQuire:STOPAfter is set to SE- Quence)
Hard copy output	HARDCopy START
Signal path compensation	*CAL? or CALibrate

Examples

BUSY?

might return 1, indicating that the oscilloscope is busy.

Related Commands

*OPC, *WAI

***CAL? (Query Only)**

Instructs the instrument to perform an internal signal path compensation and return its status.

***NOTE.** The signal path compensation can take up to a minute to respond. No other commands will be executed until signal path compensation is complete.*

Group

Calibration and Diagnostic

Syntax

*CAL?

Returns

<NR1> = 0 indicates that the signal path compensation completed without any errors detected.

<NR1> \neq 0 indicates that the signal path compensation did not complete successfully or completed with errors.

Examples

*CAL?

performs a signal path compensation and might return 0 to indicate that it completed successfully.

Related Commands

None

CALibrate:CONTINUE (No Query Form)

The CALibrate:CONTINUE command instructs the instrument to move to the next step in the oscilloscope factory calibration process.

Group

Calibration and Diagnostic

Syntax

CALibrate:CONTINUE

Arguments

None

Examples

None

Related Commands

CALibrate:FACTory

CALibrate:FACTory (No Query Form)

The CALibrate:FACTory command starts the oscilloscope factory calibration process.

The calibration process consists of a series of steps. You proceed from one step to the next by sending the CALibrate:CONTINUE command. The calibration process automatically sets up the instrument for each step. Do not send any other commands to change the state of the instrument until the calibration process is complete. You can send *RST to abort the calibration process.

Group

Calibration and Diagnostic

Syntax

CALibrate:FACTory

Arguments

None

Examples

None

Related Commands

CALibrate:CONTINUE

CALibrate:INTERNAL (No Query Form)

Instructs the instrument to perform an internal signal path compensation and return its status.

NOTE. *The signal path compensation can take up to a minute to respond. No other commands will be executed until signal path compensation is complete.*

Group

Calibration and Diagnostic

Syntax

CALibrate:INTERNAL

Returns

<NR1> = 0 indicates that the signal path compensation completed without any errors detected.

<NR1> \neq 0 indicates that the signal path compensation did not complete successfully or completed with errors.

Examples

CALibrate:INTERNAL
performs a signal path compensation and might return 0 to indicate that it completed successfully.

Related Commands

None

CALibrate:STATUS?

The CALibrate:STATUS? query returns the status from the last adjustment sequence.

Group

Calibration and Diagnostic

Syntax

CALibrate:STATUS?

Returns

PASS indicates that the last adjustment sequence completed without any errors detected.

FAIL indicates that the last adjustment sequence did not complete successfully or completed with errors.

Examples

CALibrate:STATUS?

might return :CALIBRATE:STATUS FAIL if oscilloscope calibration did not pass.

Related Commands

CALibrate:FACTory

CH<x>? (Query Only)

Returns the oscilloscope vertical parameters. Because CH<x>:SCALE and CH<x>:VOLts are identical, only CH<x>:SCALE is returned.

Group

Vertical

Syntax

CH<x>?

Returns

None

Examples

CH1?

might return the string :CH1:SCALE 1.0E0;POSITION 0.0E0;
OFFSET 0.0E0;COUPLING DC;BANDWIDTH FULL;
PROBE:UNITS VOLTS;VOLTSCALE 1.0E1;CURRENTSCALE 1.0E-3;
:CH1:INVERT 0 for channel 1.

Related Commands

None

CH<x>:BANDwidth

Sets or queries the bandwidth setting of the specified oscilloscope channel. This command is equivalent to setting Bandwidth in the Vertical menu.

Group

Vertical

Syntax`CH<x>:BANDwidth { TWEnty | FULL }``CH<x>:BANDwidth?`**Arguments**

TWEnty sets the channel bandwidth to 20 MHz.

FULL sets the channel bandwidth to the full bandwidth of the instrument.

Examples

`CH2:BANDWIDTH TWENTY`
sets the bandwidth of channel 2 to 20 MHz.

`CH1:BANDWIDTH?`
might return FULL, which indicates that there is no bandwidth limiting on channel 1.

Related Commands

None

CH<x>:COUPLing

Sets or queries the input attenuator coupling setting of the specified oscilloscope channel. This command is equivalent to setting Coupling in the Vertical menu.

Group

Vertical

Syntax

```
CH<x>:COUPLing { AC | DC | GND }
```

```
CH<x>:COUPLing?
```

Arguments

AC sets the specified oscilloscope channel to AC coupling.

DC sets the specified oscilloscope channel to DC coupling.

GND sets the specified oscilloscope channel to ground. Only a flat ground-level waveform is displayed.

Examples

```
CH1:COUPLING AC  
establishes AC coupling on channel 1.
```

```
CH2:COUPLING?  
might return DC, indicating that channel 2 is set to DC coupling.
```

Related Commands

None

CH<x>:INVert

Sets or queries the input polarity setting of the specified oscilloscope channel. This command is equivalent to setting Invert in the Vertical menu.

Group

Vertical

Syntax

CH<x>:INVert { ON | OFF | <NR1> }

CH<x>:INVert?

Arguments

<NR1> = 0 sets the specified oscilloscope channel to invert off (normal display).

<NR1> = 1 sets the specified oscilloscope channel to invert on.

Examples

CH1:INVert OFF

establishes a non-inverted display on channel 1.

CH2:INVert?

might return 1, indicating that channel 2 is inverted.

Related Commands

None

CH<x>:POStion

Sets or queries the vertical position of the specified oscilloscope channel. The position voltage value is applied to the signal before digitization. This command is equivalent to setting Position in the Vertical menu or adjusting the front-panel VERTICAL POSITION rocker.

Group

Vertical

Syntax

CH<x>:POStion <NR3>

CH<x>:POStion?

Arguments

<NR3> is the desired position, in divisions from the center graticule. The range is ± 5 divisions.

Examples

CH2:POSITION 1.3E+00

positions the channel 2 input signal 1.3 divisions above the center of the display.

CH1:POSITION?

might return $-1.3E+00$, indicating that the current position of channel 1 is at -1.3 divisions.

Related Commands

None

CH<x>:PRObe? (Query Only)

Returns the voltage probe and current probe scale factors and units for the specified oscilloscope channel.

Group

Vertical

Syntax

CH<x>:PRObe?

Returns

<NR3>

Examples

CH2:PROBE?

might return :CH2:PROBE:UNITS VOLTS;VOLTSCALE 1.0E1;
CURRENTSCALE 1.0E-3.

Related Commands

None

CH<x>:PROBe:CURrentscale

Sets or queries the conversion factor of the current probe that is attached to the specified oscilloscope channel.

Group

Vertical

Syntax

CH<x>:PROBe:CURrentscale <NR3>

CH<x>:PROBe:CURrentscale?

Arguments

<NR3> = 1.0E-3 sets the conversion factor for a 1 mV/A current probe.

<NR3> = 1.0E-2 sets the conversion factor for a 10 mV/A current probe.

<NR3> = 1.0E-1 sets the conversion factor for a 100 mV/A current probe.

<NR3> = 1.0E0 sets the conversion factor for a 1 V/A current probe.

Examples

CH<x>:PROBe:CURrentscale 0.1
sets the scale for a 100 mV/A current probe.

Related Commands

CH<x>:PROBe:UNITs

CH<x>:PROBe:UNITs

Sets or queries the probe type, voltage or current, that is attached to the specified oscilloscope channel.

Group

Vertical

Syntax

CH<x>:PROBe:UNITs { VOLts | AMPS }

CH<x>:PROBe:UNITs?

Arguments

<VOLts> sets the oscilloscope probe to a voltage probe.

<AMPS> sets the oscilloscope probe to a current probe .

Examples

CH<x>:PROBe:UNITs AMPS

sets the oscilloscope probe to be a current probe.

Related Commands

CH<x>:PROBe:SCALe

CH<x>:PROBe:VOLTSCALE

Sets or queries the attenuation factor of the voltage probe that is attached to the specified oscilloscope channel.

Group

Vertical

Syntax

CH<x>:PROBe:VOLTSCALE <NR3>

CH<x>:PROBe:VOLTSCALE?

Arguments

<NR3> = 1.0E0 sets the attenuation factor for a 1X probe.

<NR3> = 1.0E1 sets the attenuation factor for a 10X probe.

<NR3> = 1.0E2 sets the attenuation factor for a 100X probe.

<NR3> = 1.0E3 sets the attenuation factor for a 1000X probe.

Examples

CH<x>:PROBe:VOLTSCALE 100
sets the scale for a 100X voltage probe.

Related Commands

CH<x>:PROBe:UNITs

CH<x>:SCAlE

Sets or queries the vertical gain of the specified oscilloscope channel. This command is equivalent to adjusting the front-panel VOLTS/DIV rocker.

Group

Vertical

Syntax

CH<x>:SCAlE <NR3>

CH<x>:SCAlE?

Arguments

<NR3> is the gain, in volts per division. The range is 10 V/div to 2 mV/div when using a 1X probe.

Examples

CH1:SCALE 100E-03
sets the channel 1 gain to 100 mV/div.

CH2:SCALE?
might return 1.0E0, indicating that the current V/div setting of channel 2 is 1 V/div.

Related Commands

CH1:VOLts

CH<x>:VOLts

Sets or queries the vertical gain of the specified oscilloscope channel. This command is identical to the CH<x>:SCALE command and is included for compatibility purposes. Only CH<x>:SCALE is returned in response to a CH<x>? query.

Group

Vertical

Syntax

CH<x>:VOLts <NR3>

CH<x>:VOLts?

Arguments

None

Examples

CH1:VOLTS 100E-3
sets the channel 1 gain to 100 mV/div.

CH2:VOLTS?
might return 1.0E0, indicating that the current V/div setting of channel 2 is 1 V/div.

Related Commands

CH1:SCALE

CLEARMenu (No Query Form)

Clears the current menu from the display. This command is equivalent to pressing the CLEAR MENU button on the front panel.

Group

Display

Syntax

CLEARMenu

Arguments

None

Examples

CLEARMENU

clears the menu from the display.

Related Commands

None

***CLS (No Query Form)**

Clears the instrument status data structures.

Group

Status and Error

Syntax

*CLS

The *CLS command clears the following:

- the Event Queue
- the Standard Event Status Register (SESR)
- the Status Byte Register (except the MAV bit; see below)

If the *CLS command immediately follows a <LF>, the Output Queue and MAV bit (Status Byte Register bit 4) are also cleared. MAV indicates information is in the output queue. (A complete discussion of these registers and bits, and of event handling in general, begins on page 3–1.)

Arguments

None

Examples

None

Related Commands

DESE, *ESE, *ESR?, EVENT?, EVMsg?, *SRE, *STB?

CURSor? (Query Only)

Returns all current oscilloscope cursor settings.

Group

Cursor

Syntax

CURSor?

Returns

None

Examples

CURSOR?

might return :CURSOR:FUNCTION OFF;VBARS:UNITS SE-
CONDS;POSITION1 500.0E-6;POSITION2 4.50E-3;SELECT
CURSOR1;:CURSOR:HBARS:POSITION1 3.20E+0;POSITION2
-3.20E+0;SELECT CURSOR1 as the current cursor settings.

Related Commands

None

CURSor:FUNction

Selects and displays the oscilloscope cursor type. Cursors are attached to the selected channel. This command is equivalent to setting Function in the Cursor menu.

Group

Cursor

Syntax

CURSor:FUNction { HBArS | OFF | PAIred | VBArS }

CURSor:FUNction?

Arguments

HBArS specifies horizontal bar cursors that measure the vertical units (volts, amperes, watts).

OFF removes the cursors from the display.

VBArS specifies vertical bar cursors that measure time or frequency.

PAIred specifies paired cursors that show both horizontal and vertical units.

Examples

CURSOR:FUNction VBARS
selects vertical bar type cursors.

Related Commands

SElect:CONTROL, DMM:CURSor:FUNction

CURSor:HBArS? (Query Only)

Returns the current settings for the oscilloscope horizontal bar cursors.

Group

Cursor

Syntax

CURSor:HBArS?

Returns

None

Examples

CURSOR:HBARS?
might return :CURSOR:HBARS:POSITION1 0;POSITION2 0;SELECT
CURSOR1.

Related Commands

DMM:CURSor:HBArS?

CURSor:HBARs:DELTA? (Query Only)

Returns the difference (in vertical units) between the two horizontal bar cursors in the oscilloscope display.

Group

Cursor

Syntax

CURSor:HBARs:DELTA?

Returns

<NR3>

Examples

CURSOR:HBARS:DELTA?

might return 5.08E+0 for the difference between the two cursors.

Related Commands

DMM:CURSor:HBARs:DELTA?

CURSor:HBARs:POSITION<x>

Positions an oscilloscope horizontal bar cursor.

Group

Cursor

Syntax

CURSor:HBARs:POSITION<x> <NR3>

CURSor:HBARs:POSITION<x>?

Arguments

<NR3> specifies the cursor position relative to ground, in vertical units.

Examples

```
CURSOR:HBARS:POSITION1 25.0E-3
```

positions one of the horizontal cursors at 25.0 mV (assuming the vertical units are volts).

```
CURSOR:HBARS:POSITION2?
```

might return $-64.0E-3$, indicating that one of the horizontal bar cursors is at -64.0 mA (assuming the vertical units are amperes).

Related Commands

```
DMM:CURSor:HBARs:POSITION<x>
```

CURSor:HBARs:SElect

Selects which oscilloscope horizontal bar cursor is active for front-panel control. This command is equivalent to pressing the TOGGLE button on the front panel. The active cursor is displayed as a solid horizontal line and can be moved using the front-panel +/- rocker. The unselected cursor is displayed as a dashed horizontal line.

Group

Cursor

Syntax

```
CURSor:HBARs:SElect { CURSOR1 | CURSOR2 }
```

```
CURSor:HBARs:SElect?
```

Arguments

CURSOR1 selects the first horizontal bar cursor.

CURSOR2 selects the second horizontal bar cursor.

Examples

```
CURSOR:HBARS:SELECT CURSOR1
```

selects the first horizontal bar cursor as the active cursor.

```
CURSOR:HBARS:SELECT?
```

returns CURSOR1 when the first cursor is the active cursor.

Related Commands

DMM:CURSor:HBArs:SElect

CURSor:PAIred? (Query Only)

Queries and returns the oscilloscope paired cursor measurements.

Group

Cursor

Syntax

CURSor:PAIred?

Returns

None

Examples

CURSOR:PAIred?

might return :CURSOR:PAIRED:POSITION1 0;POSITION2 0;SELECT
CURSOR1.

Related Commands

DMM:CURSor:PAIred?

CURSor:PAIred:HDELta? (Query Only)

Queries the vertical distance between the first and second paired cursor in the oscilloscope display. The response is the absolute value of the difference between the vertical positions of the first and second cursors.

Group

Cursor

Syntax

CURSor:PAIred:HDELta?

Returns

None

Examples

CURSOR:PAIRED:HDELTA?

might return 5.08E+0 for the vertical difference between the two cursors.

Related Commands

CURSOR:FUNCTION, DMM:CURSOR:PAIRED:HDELTA?

CURSOR:PAIRED:HPOS1? (Query Only)

Queries the vertical position of the first paired cursor in the oscilloscope display.

Group

Cursor

Syntax

CURSOR:PAIRED:HPOS1?

Returns

None

Examples

CURSOR:PAIRED:HPOS1?

might return -64.0E-3, indicating that the first cursor is at -64.0 mV.

Related Commands

CURSOR:FUNCTION, DMM:CURSOR:PAIRED:HPOS1?

CURSor:PAIred:HPOS2? (Query Only)

Queries the vertical position of the second paired cursor in the oscilloscope display.

Group

Cursor

Syntax

CURSor:PAIred:HPOS2?

Returns

None

Examples

CURSOR:PAIRED:HPOS2?

might return $-64.0E-3$, indicating the second cursor is at -64.0 mV.

Related Commands

CURSor:FUNcTion, DMM:CURSor:PAIred:HPOS2?

CURSor:PAIred:POSITION1

Sets or queries the horizontal (time or frequency) position of the first paired cursor.

Group

Cursor

Syntax

CURSor:PAIred:POSITION1 < NR3 >

CURSor:PAIred:POSITION1?

Arguments

<NR3> specifies the position of the first paired cursor.

Examples

CURSOR:PAIRED:POSITION1 9.00E-6
specifies the first paired cursor is at 9 μ s.

CURSOR:POSITION1?
might return 1.00E-6, indicating that the first paired cursor is at 1 μ s.

Related Commands

CURSOR:FUNCTION

CURSOR:PAIRED:POSITION2

Sets or queries the horizontal (time or frequency) position of the second paired cursor.

Group

Cursor

Syntax

CURSOR:PAIRED:POSITION2 < NR3 >

CURSOR:PAIRED:POSITION2?

Arguments

<NR3> specifies the position of the second paired cursor.

Examples

CURSOR:POSITION2?
might return 1.00E-6, indicating that the second paired cursor is at 1 μ s.

Related Commands

CURSOR:FUNCTION

CURSor:PAIred:SElect

Selects the active paired cursor in the oscilloscope display. This command is equivalent to pressing the TOGGLE button on the front panel. The active cursor appears as a solid vertical line. The unselected cursor appears as a dashed vertical line.

Group

Cursor

Syntax

```
CURSor:PAIred:SElect { CURSOR1 | CURSOR2 }
```

```
CURSor:PAIred:SElect?
```

Arguments

CURSOR1 specifies the first paired cursor.

CURSOR2 specifies the second paired cursor.

Examples

```
CURSOR:PAIRED:SELECT CURSOR2
```

selects the second paired cursor as the active cursor.

```
CURSOR:PAIRED:SELECT?
```

returns CURSOR1 when the first paired cursor is the active cursor.

Related Commands

```
DMM:CURSor:PAIred:SElect
```

CURSor:PAIred:VDELta? (Query Only)

Queries the horizontal (time or frequency) distance between paired cursors in the oscilloscope display.

Group

Cursor

Syntax

CURSOR:PAIred:VDELta?

Returns

None

Examples

CURSOR:PAIRED:VDELTA?

might return 1.064E+00, indicating that the time between the paired cursors is 1.064 seconds.

Related Commands

CURSOR:FUNCTION, DMM:CURSOR:PAIRED:VDELta?

CURSOR:VBARS

Positions the oscilloscope vertical bar cursors and CURSOR:VBARS? returns the current vertical bar cursor settings for horizontal position, delta, cursor selection, and units.

Group

Cursor

Syntax

CURSOR:VBARS SNAP

CURSOR:VBARS?

Arguments

SNAP positions the vertical bar cursors at DATA:START and DATA:STOP.

Examples

CURSOR:VBARS SNAP

specifies that the cursors' positions are the same as the current DATA:START and DATA:STOP values.

CURSOR:VBARS?

might return :CURSOR:VBARS:UNITS SECONDS;POSITION1

1.00E-6;
POSITION2 9.00E-6;SELECT CURSOR2.

Related Commands

DATA:START, DATA:STOP, DMM:CURSOR:VBARS?

CURSOR:VBARS:DELTA? (Query Only)

Returns the time or frequency between the two vertical bar cursors in the oscilloscope display. The units (seconds or Hertz) are specified by the CURSOR:VBARS:UNITS command.

Group

Cursor

Syntax

CURSOR:VBARS:DELTA?

Returns

<NR3>

Examples

CURSOR:VBARS:DELTA?

might return 1.064E+00, indicating that the time between the vertical bar cursors is 1.064 seconds.

Related Commands

CURSOR:VBARS:UNITS, DMM:CURSOR:VBARS:DELTA?

CURSOR:VBARS:POSITION<x>

Positions a vertical bar cursor for both vertical bar and paired cursors. The unit is specified by the CURSOR:VBARS:UNITS command.

Group

Cursor

Syntax

```
CURSor:VBARS:POSITION<x> <NR3>
```

```
CURSor:VBARS:POSITION<x>?
```

Arguments

<NR3> specifies the cursor position in the units specified by the CURSor:VBARS:UNIts command. The position is relative to the trigger position.

Examples

```
CURSOR:VBARS:POSITION2 9.00E-6
```

positions one of the vertical bar cursors at 9 μ s.

```
CURSOR:VBARS:POSITION1?
```

might return 1.00E-6, indicating a vertical bar cursors is at 1 μ s.

Related Commands

```
CURSor:VBARS:UNIts, DMM:CURSor:VBARS:POSITION<x>
```

CURSor:VBARS:SElect

Selects which oscilloscope vertical bar cursor is active. This command is equivalent to pressing the TOGGLE button on the front panel. The active cursor is displayed as a solid vertical line and is moved using the front-panel +/- rocker. The unselected cursor is displayed as a dashed vertical line.

Group

Cursor

Syntax

```
CURSor:VBARS:SElect { CURSOR1 | CURSOR2 }
```

```
CURSor:VBARS:SElect?
```

Arguments

CURSOR1 specifies the first vertical bar cursor.

CURSOR2 specifies the second vertical bar cursor.

Examples

`CURSOR:VBARS:SELECT CURSOR2`

selects the second vertical bar cursor as the active cursor.

`CURSOR:VBARS:SELECT?`

returns CURSOR1 when the first vertical bar cursor is the active cursor.

Related Commands

`DMM:CURSor:VBARS:SElect`

CURSor:VBArS:UNItS

Sets or queries the units for the oscilloscope vertical bar cursors. This is equivalent to setting Time Units in the Cursor menu.

Group

Cursor

Syntax

`CURSor:VBArS:UNItS { SECOnds | HERTz }`

`CURSor:VBArS:UNItS?`

Arguments

None

Examples

`CURSOR:VBARS:UNITS SECONDS`

sets the units for the vertical bar cursors to seconds.

`CURSOR:VBARS:UNITS?`

returns HERTZ when the vertical bar cursor units are Hertz.

Related Commands

`CURSor:VBArS:DELTA?`, `CURSor:VBArS:POSITION<x>`

CURVe

Transfers oscilloscope waveform data to and from the instrument in binary or ASCII format. Each waveform that is transferred has an associated waveform preamble that contains information such as data format and scale. Refer to the WFMPre command starting on page 2–239 for information about the waveform preamble. The data format is specified by the DATA:ENCdg and DATA:WIDTH commands.

The CURVe? query transfers data from the instrument. The data source is specified by the DATA:SOURce command. If more than one source is specified, a comma-separated list of data blocks is returned. The first and last data points that are transferred are specified by the DATA:STARt and DATA:STOP commands.

The CURVe command transfers waveform data to the instrument. The data is stored in the stored waveform location specified by DATA:DESTination starting with the data point specified by DATA:STARt. Only one waveform can be transferred at a time. To display a transferred waveform, map the destination stored waveform to one of the reference waveforms and display that reference waveform (See REFA and REFB commands).

A description of the waveform transfer process starts on page 2–26.

Group

Waveform

Syntax

```
CURVe { <Block> | <asc curve> }
```

CURVe?

Arguments

<Block> is the waveform data in binary format. The waveform is formatted as: #<x><yyy><data><newline> where <x> is the number of y bytes. For example, if <yyy> = 500, then <x> = 3. <yyy> is the number of bytes to transfer. If width is 1, then all bytes on the bus are single data points. If width is 2, then all bytes on the bus are 2-byte pairs. Use the DATA:WIDth command to set the width. <data> is the curve data. <newline> is a single byte newline character at the end of

the data. See the GETWFM.C or GETWFM.BAS examples in the accompanying disk for more specifics.

<asc curve> is the waveform data in ASCII format. The format for ASCII data is <NR1>[,<NR1>...] where each <NR1> represents a data point.

Examples

CURVE?

might return the ASCII data

```
:CURVE 13,6,3,2,-1,-9,-14,-19,-29,-35,-67,-71,-78,-62,-50,-31,-27,-7,0,12,29,39,45,43,41,47,41,38,33,26
```

Related Commands

DATA, WFMPre

DATA

Sets or queries the format and location of the oscilloscope waveform data that is transferred with the CURVE command. Since DATA:DESTINATION and DATA:TARGET are equivalent, only DATA:DESTINATION is returned by the DATA? query.

Group

Waveform

Syntax

```
DATA { INIT | SNAp }
```

DATA?

Arguments

INIT initializes the waveform data parameters to their factory defaults.

SNAp sets DATA:START and DATA:STOP to match the current vertical bar cursor positions.

Examples

DATA SNAP

assigns DATA:START and DATA:STOP to the current position of the vertical bar cursors.

DATA?

might return the string :DATA:ENCDG RPBINARY;DESTINATION ST04; SOURCE REF2;START 1;STOP 500;WIDTH 2

Related Commands

CURVE, WAVFrm

DATA:DESTINATION

Sets or queries the reference memory location for storing oscilloscope waveform data that is transferred into the instrument by the CURVE command. This command is identical to the DATA:TARGET command.

Group

Waveform

Syntax

DATA:DESTINATION { REFA | REFB | ST0[1-10] }

DATA:DESTINATION?

Arguments

REFA waveform is stored into the location currently mapped to REFA.

REFB waveform is stored into the location currently mapped to REFB.

ST0[1-10] waveform is stored into one of ten waveform storage locations.

Examples

DATA:DESTINATION REFA

stores incoming waveform data into the stored waveform currently mapped to Ref A.

DATA:DESTINATION?

might return REFB as the waveform location that is currently selected.

Related Commands

None

DATA:ENCdg

Sets or queries the format of the oscilloscope waveform data. This command is equivalent to setting WFMPre:ENCdg, WFMPre:BN_Fmt, and WFMPre:BYT_Or as shown in Table 2–28. Setting the DATA:ENCdg value causes the corresponding WFMPre values to be updated and vice versa.

Group

Waveform

Syntax

```
DATA:ENCdg { ASCIIi | RIBinary | RPBinary | SRIBinary | SRPbinary }
```

DATA:ENCdg?

Arguments

ASCIIi specifies the ASCII representation of signed integer (RIBinary) data. If this is the value at power-on, the WFMPre values for BN_Fmt, BYT_Or, and ENCdg are set as RP, MSB, and ASC respectively.

RIBinary specifies signed integer data-point representation with the most significant byte transferred first. This format results in the fastest data transfer rate when DATA:WIDTH is set to 2.

The range is –128 to 127 when DATA:WIDTH is 1. Zero is center screen. The range is –32768 to 32767 when DATA:WIDTH is 2. The upper limit is one division above the top of the screen and the lower limit is one division below the bottom of the screen.

RPBinary specifies positive integer data-point representation with the most significant byte transferred first.

The range is 0 to 255 when DATA:WIDTH is 1. Center screen is 127. The range is 0 to 65,535 when DATA:WIDTH is 2. The upper limit is one division above the top of the screen and the lower limit is one division below 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 IBM compatible PCs.

Table 2–28: DATA and WFMPre Parameter Settings

DATA:ENCdg Setting	WFMPre Settings		
	:ENCdg	:BN_Fmt	:BYT_Or
ASCIi	ASC	N/A	N/A
RIBinary	BIN	RI	MSB
RPBinary	BIN	RP	MSB
SRIBinary	BIN	RI	LSB
SRPbinary	BIN	RP	LSB

Examples

DATA:ENCDG RPBINARY

sets the data encoding format to be positive integer where the most significant byte is transferred first.

DATA:ENCDG?

might return SRPBINARY for the format of the waveform data.

Related Commands

WFMPre:ENCdg, WFMPre:BN.FMT, WFMPre:BYT_Or

DATA:SOURce

Sets or queries the location of the oscilloscope waveform data that is transferred from the instrument by CURVE?. The source data is always transferred in a predefined order regardless of the order they are specified using this command. The predefined order is CH1, CH2, MATH, REFA, REFB, and then stored waveforms 1 through 10 (STO[1–10]).

Group

Waveform

Syntax

DATA:SOURce <wfm> [<Comma><wfm>] . . .

DATA:SOURce?

Arguments

<wfm> is the location of the waveform data that will be transferred from the instrument to the controller.

Examples

DATA:SOURCE REFB, CH2, MATH1, CH1, ST08 specifies that five waveforms will be transferred in the next CURVE? query. The order that the data will be transferred is CH1, CH2, MATH, then REFB, and then stored waveform 8.

DATA:SOURCE?

might return REFA, indicating the source for the waveform data that is transferred using CURVE?.

Related Commands

None

DATA:START

Sets or queries the starting data point for oscilloscope waveform and DMM logger data transfers. This command allows for the transfer of partial waveforms to and from the instrument (transfer from the instrument only for DMM logger data).

Group

Waveform

Syntax

DATA:START <NR1>

DATA:START?

Arguments

<NR1> ranges from 1 to 2500 for oscilloscope waveforms. <NR1> ranges from 1 to 480 for DMM logger data. This value is the first data point that will be transferred. Data is transferred from <NR1> to DATA:STOP or the record length, whichever is less. When DATA:STOP is less than DATA:START, the values are swapped internally for CURVe?.

Examples

DATA:START 10

specifies that the waveform transfer will begin with data point 10.

DATA:START?

might return 214 as the first waveform data point that will be transferred.

Related Commands

CURVe?, DATA:SNAP, DATA:STOP

DATA:STOP

Sets or queries the last data point in the oscilloscope waveform or DMM logger data that will be transferred when using CURVe?. This allows the transfer of partial waveforms to the controller.

When using the CURVe command, the instrument stops reading data when there is no more data to read or when the specified record length is reached; this command is ignored.

Group

Waveform

Syntax

DATA:STOP <NR1>

DATA:STOP?

Arguments

<NR1> ranges from 1 to 2500 for oscilloscope waveforms. <NR1> ranges from 1 to 480 for DMM logger data. This value is the last data point that will be transferred. When DATA:STOP is less than DATA:START, the values are swapped internally for CURVe?.

If you always want to transfer complete waveforms, set DATA:START to 1 and DATA:STOP to the record length (2500).

Examples

DATA:STOP 150

specifies that the waveform transfer will stop at data point 150.

DATA:STOP?

might return 285 as the last data point that will be transferred.

Related Commands

CURVe?, DATA:SNAP

DATA:TARget

Sets or queries the location for storing oscilloscope waveform data transferred to the instrument using the CURVe command. This command is equivalent to the DATA:DESTINATION command and is included here for compatibility with older Tektronix instruments.

Group

Waveform

Syntax

DATA:TARget { REFA | REFB | ST0[1-10] }

DATA:TARget?

Arguments

REFA waveform is stored into the location currently mapped to REFA.

REFB waveform is stored into the location currently mapped to REFB.

ST0[1-10] waveform is stored into one of ten waveform storage locations.

Examples

DATA:TARget REFA

stores incoming waveform data into the stored waveform currently mapped to Ref A.

DATA:TARget?

might return REFB as the waveform location that is currently selected.

Arguments

None

Examples

None

Related Commands

CURVe

DATA:WIDTH

Sets the number of bytes per data point in the oscilloscope waveform or DMM logger data transferred using the CURVe command.

Group

Waveform

Syntax

DATA:WIDTH <NR1>

DATA:WIDTH?

Arguments

<NR1> = 1 specifies that there is 1 byte (8 bits) per point. This format is useful when the acquisition mode is set to SAMple, ENvelope, or PEAKdetect. If used for AVErage, the low order byte is not transmitted.

<NR1> = 2 specifies that there are 2 bytes (16 bits) per point. This format is useful for AVErage waveforms. If used for ENvelope, PEAKdetect, or SAMple, the least significant byte is always zero.

If DATA:WIDTH is set to 2, the block is twice as long as when it is 1. The length or number of bytes in the block can be calculated by $((\text{DATA:STOP} - \text{DATA:START}) + 1) * \text{DATA:WIDTH}$. If DATA:START and/or DATA:STOP extend beyond the limits of the waveform the number of bytes will be less.

Examples

DATA:WIDTH 1

sets the data width to 1 byte per data point for CURVe data.

Related Commands

CURVe, WFMPre:BIT_Nr, WFMPre:BYT_Nr

*DDT

Allows the user to specify a command or a list of commands that are executed when the instrument receives a *TRG command. This is a special alias that *TRG uses.

Group

Miscellaneous Oscilloscope

Syntax

*DDT { <Block> | <QString> }

*DDT?

Arguments

<Block> or <QString> is a complete sequence of program messages. The messages must contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands (see page 2–4). The sequence must be ≤80 characters. <Block> format is always returned as a query response.

Examples

```
*DDT #217ACQUIRE:STATE RUN<EOI>
```

specifies that the acquisition system will be started each time a *TRG command is sent.

Related Commands

ALIAS:DEFINE, *TRG

DESE

Sets and queries the bits in the Device Event Status Enable Register (DESER). The DESER is the mask that determines whether or not events are reported to the Standard Event Status Register (SESR), and entered into the Event Queue. For a more detailed discussion of the use of these registers, see page 3–1.

Group

Status and Error

Syntax

```
DESE <NR1>
```

```
DESE?
```

Arguments

<NR1> is a value in the range from 0 to 255. The binary bits of DESER are set according to this value. 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, and so on).

The power-on default for DESER is all bits set if *PSC is 1. If *PSC is 0, the DESER maintains its value through a power cycle.

NOTE. Setting *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 *ESER*. A discussion of event handling begins on page 3–1.

Examples

DESE 209

sets the *DESER* to binary 11010001, which enables the *PON*, *URQ*, *EXE*, and *OPC* bits.

DESE?

might return the string :DESE 186, showing that *DESER* contains the binary value 10111010.

Related Commands

*CLS, *ESE, *ESR?, EVENT?, EVMsg?, *SRE, *STB?

DIAG:RESULT:FLAG? (Query Only)

Returns the pass/fail status from the last diagnostic test sequence execution. Used the *DIAG:RESULT:LOG?* query to determine which test(s) has failed.

Group

Calibration and Diagnostic

Syntax

DIAG:RESULT:FLAG?

Returns

PASS indicating that all of the selected diagnostic tests have passed.

FAIL indicating that at least one of the selected diagnostic tests have failed.

Examples

DIAG:RESULT:FLAG?

returns either PASS or FAIL.

Related Commands

DIAG:RESULT:LOG?

DIAG:RESULT:LOG? (Query Only)

Returns the internal results log from the last diagnostic 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?

Returns

<QString> in the following format:

<Status>,<Module name>[,<Status>,<Module name>...]

Examples

DIAG:RESULT:LOG?

might return :DIAG:RESULT:LOG "pass--Processor,pass--Display, pass--FP/Proc Interface,fail--Front Panel"

Related Commands

DIAG:RESULT:FLAG?

DIAG:SElect:ALL (No Query Form)

Specifies that all system test sequences will be run when the DIAG:STATE EXECUTE command is sent.

Group

Calibration and Diagnostic

Syntax

DIAG:SElect:ALL ALL

Arguments

ALL selects functional, memory, and register tests for the acquisition, processor and display systems, and self diagnostics for the front panel.

Examples

None

Related Commands

None

DIAG:STATE EXECute (No Query Form)

Executes the diagnostic tests specified by the DIAG:SElect command.

When the test sequence has completed, any of the modules or module interfaces that failed diagnostics are displayed on the screen and stored in an internal log file. The pass/fail status is returned by DIAG:RESUlt:FLAg? and the internal log is returned by DIAG:RESUlt:LOG?. This command is equivalent to running Extended Diagnostics by selecting Execute in the Utility menu when System is set to Diag.

NOTE. *The DIAG:STATE EXECute command can take 30 seconds or more to respond. This command does not return control to the instrument controller until diagnostics are complete.*

Group

Calibration and Diagnostic

Syntax

DIAG:STATE EXECute

Arguments

EXECute runs the diagnostic test sequences specified by the DIAG:SElect command. When complete, the instrument returns to

the state it was in just prior to the test. The pass/fail status of the tests can be returned by executing `DIAG:RESULT:FLAG?`.

Examples

`DIAG:STATE EXECUTE`

executes all the diagnostic tests that have been selected.

Related Commands

`DIAG:RESULT:FLAG?`, `DIAG:RESULT:LOG?`

DISplay? (Query Only)

Returns the current display settings.

Group

Display

Syntax

`DISplay?`

Returns

None

Examples

`DISPLAY?`

might return `:DISPLAY:MODE SCOPE;FORMAT YT;STYLE VEC-TORS;PERSISTENCE 500.0E-3;GRATICULE FULL;TRIGT 1;CON-TRAST 50`

Related Commands

None

DISplay:CONTRast

Sets or queries the contrast of the display.

Group

Display

Syntax

DISplay:CONTRast <NR1>

DISplay:CONTRast?

Arguments

<NR1> is an integer in the range from 1 through 100.

Examples

DISplay:CONTRast 63
sets the display contrast to 63.

Related Commands

None

DISplay:FORMat

Sets or queries the oscilloscope display format. This command is equivalent to setting Format in the oscilloscope Display menu.

Group

Display

Syntax

DISplay:FORMat { XY | YT }

DISplay:FORMat?

Arguments

XY displays the vertical values of one waveform against the vertical values of another. The sources that make up an XY waveform are

predefined and are listed in Table 2–29. Displaying one source causes its corresponding source to be displayed.

Table 2–29: XY Format Pairs

X-Axis Source	Y-Axis Source
Ch 1	Ch 2
RefA	RefB

YT sets the display to vertical values versus time format and is the normal mode.

Examples

`DISPLAY:FORMAT YT`

selects a voltage, amperes, or watts versus time format for the display.

`DISPLAY:FORMAT?`

might return XY for the display format.

Related Commands

None

DISplay:GRAticule

Selects the type of oscilloscope graticule that is displayed. This command is equivalent to setting Graticule in the Display menu.

Group

Display

Syntax

`DISplay:GRAticule { FRAmE | FULL | GRID | CROSS }`

`DISplay:GRAticule?`

Arguments

FRAMe specifies just a frame.

FUL1 specifies a frame, a grid, and cross hairs.

GRId specifies a frame and grid only.

CROss specifies a frame and cross hairs only.

Examples

DISPLAY:GRATICULE FRAME

sets the graticule type to display the frame only.

DISPLAY:GRATICULE?

returns FULL when all graticule elements (grid, frame, and cross hairs) are selected.

Related Commands

None

DISplay:MODE

Selects the functional mode of the instrument: scope mode or meter mode. This command is equivalent to pressing the front-panel SCOPE or METER buttons.

Group

Display

Syntax

DISplay:MODE { METer | SCOpe }

DISplay:MODE?

Arguments

METer specifies meter mode.

SCOpe specifies scope mode.

Examples

DISPLAY:MODE SCOpe

sets instrument to operate on scope mode.

DISPLAY:MODE?

returns METER if the instrument is in meter mode.

Related Commands

None

DISplay:PERsistence

Sets the length of time that dots (or vectors) are displayed when the oscilloscope DISplay:STYle is set to ACCUMDOTS or ACCUM-VECTORS.

Group

Display

Syntax

DISplay:PERsistence { <NR3> }

DISplay:PERsistence?

Arguments

<NR3> specifies the length, in seconds, that the waveform points are displayed on the screen. The range is 100 ms to 10 s.

NR3 = 0 specifies infinite persistence.

Examples

DISPLAY:PERSISTENCE 3

specifies that the waveform points are displayed on the screen for 3 seconds before they fade.

Related Commands

DISplay:STYle

DISplay:STYLE

Selects how the oscilloscope waveform data is displayed. This command is equivalent to setting Style in the oscilloscope Display menu.

Group

Display

Syntax

```
DISplay:STYLE { DOTS | ACCUMDots | ACCUMVectors |  
VECTors }
```

DISplay:STYLE?

Arguments

DOTs displays individual data points.

ACCUMDots accumulates data points on the display until the PERSistence time is met.

VECTors connects adjacent data points. Old points are immediately replaced by new ones.

ACCUMVectors accumulates data points with a line vector waveform until the PERSistence time is met.

Examples

```
DISPLAY:STYLE VEC  
sets the display to connect adjacent data points.
```

```
DISPLAY:STYLE?  
might return DOTS indicating that the display shows individual  
waveform data points.
```

Related Commands

DISplay:PERSistence

DISplay:TRIGT

Controls the display of the oscilloscope trigger indicator. This command is equivalent to setting the Display 'T' @ Trig Pt in the oscilloscope Display menu. The query form returns an ON (1) or an OFF (0).

Group

Display

Syntax

```
DISplay:TRIGT { OFF | ON | <NR1> }
```

```
DISplay:TRIGT?
```

Arguments

<OFF> or <NR1> = 0 removes the trigger indicator from the display.

<ON> or <NR1> \neq 0 displays a trigger indicator on each of the displayed waveforms. The trigger indicator is in reverse video for the selected waveform.

Examples

```
DISPLAY:TRIGT ON
```

sets the display to show trigger indicators.

```
DISPLAY:TRIGT?
```

might return 1 indicating that the display shows trigger indicators.

Related Commands

None

DMM:BEEP:CONTInuity

Enables or disables the audible beeper for continuity measurements.

Group

DMM Beeper

Syntax

```
DMM:BEEP:CONTInuity { OFF | ON }
```

```
DMM:BEEP:CONTInuity?
```

Arguments

<OFF> disables the beeper.

<ON> enables the beeper.

Examples

```
DMM:BEEP:CONTInuity ON
```

causes a beep when circuit continuity is detected.

Related Commands

None

DMM:BEEP:NEWMax

Enables or disables the audible beep that sounds when the Min or Max statistic is updated with a new value.

Group

DMM Beeper

Syntax

```
DMM:BEEP:NEWMax { OFF | ON }
```

```
DMM:BEEP:NEWMax?
```

Arguments

<OFF> disables the beeper for the new Max or Min value.

<ON> enables the beeper for the new Max or Min value.

Examples

DMM:BEEP:NEWMax?

returns OFF if the beep that sounds for new Max or Min values is turned off.

Related Commands

None

DMM:CALibrate:CONTINUE (No Query Form)

The DMM:CALibrate:CONTINUE command instructs the oscilloscope to move to the next step in the DMM factory calibration process.

Group

DMM Calibration

Syntax

DMM:CALibrate:CONTINUE

Arguments

None

Examples

None

Related Commands

DMM:CALibrate:FACTory

DMM:CALibrate:FACTory (No Query Form)

Issue this command to start the DMM factory calibration procedure.

The calibration process consists of a series of steps. You proceed from one step to the next by sending the DMM:CALibrate:CONTINUE command. The calibration process automatically sets up the

instrument for each step. Do not send any other commands to change the state of the instrument until the calibration process is complete. You can send *RST to abort the calibration process.

Group

DMM Calibration

Syntax

DMM:CALibrate:FACTory

Arguments

None

Examples

None

Related Commands

DMM:CALibrate:CONTINUE

DMM:CALibrate:STATUS? (Query Only)

Use this query to return the status of the last calibration step.

Group

DMM Calibration

Syntax

DMM:CALibrate:STATUS?

Returns

PASS indicates that the calibration completed without any errors detected.

FAIL indicates that the calibration did not complete successfully or completed with errors.

Examples

DMM:CALibrate:STATUS?

might return :DMM:CALIBRATE:STATUS FAIL if DMM calibration did not pass.

Related Commands

None

DMM:CURSor:FUNcTION

Selects and displays the DMM cursor type. The DMM cursors operate on the logger plot. This command is equivalent to setting Function in the DMM Cursor menu.

Group

DMM Logger

Syntax

DMM:CURSor:FUNcTION { HBArS | OFF | PAIred | VBArS }

DMM:CURSor:FUNcTION?

Arguments

HBArS specifies horizontal bar cursors that measure vertical units (volts, amperes, and ohms).

OFF removes the cursors from the display.

VBArS specifies vertical bar cursors that measure time.

PAIred specifies paired cursors that show both vertical units and time.

Examples

DMM:CURSOR:FUNcTION VBARS
selects vertical bar type cursors.

Related Commands

CURSor:FUNcTION

DMM:CURSor:HBARs? (Query Only)

Returns the current settings for the DMM horizontal bar cursors.

Group

DMM Logger

Syntax

DMM:CURSor:HBARs?

Returns

None

Examples

DMM:CURSOR:HBARS?

might return :DMM:CURSOR:HBARS:POSITION1 0;POSITION2
0;SELECT CURSOR1.

Related Commands

CURSor:HBARs?

DMM:CURSor:HBARs:DELTA? (Query Only)

Group

Returns the difference (in vertical units) between the two horizontal bar cursors in the logger plot.

Syntax

DMM:CURSor:HBARs:DELTA?

Returns

<NR3>

Examples

DMM:CURSOR:HBARS:DELTA?

might return 5.08E+0 for the voltage difference between the two cursors.

Related Commands

CURSOR:HBARS:DELTA?

DMM:CURSOR:HBARS:POSITION<x>

Positions a data logger horizontal bar cursor.

Group

DMM Logger

Syntax

DMM:CURSOR:HBARS:POSITION<x> <NR3>

DMM:CURSOR:HBARS:POSITION<x>?

Arguments

<NR3> specifies the cursor position relative to ground, in vertical units (volts, amperes, or ohms).

Examples

DMM:CURSOR:HBARS:POSITION1 25.0E-3

positions one of the horizontal cursors at 0.025 vertical units.

DMM:CURSOR:HBARS:POSITION2?

might return -64.0E-3, indicating that one of the horizontal bar cursors is at -64.0 mV.

Related Commands

CURSOR:HBARS:POSITION<x>

DMM:CURSOR:HBARs:SElect

Selects which data logger horizontal bar cursor is active for front-panel control. The active cursor is displayed as a solid horizontal line and can be moved using the front-panel +/- rocker. The unselected cursor is displayed as a dashed horizontal line. This command is equivalent to pressing the TOGGLE button on the front panel.

Group

DMM Logger

Syntax

```
DMM:CURSOR:HBARs:SElect { CURSOR1 | CURSOR2 }
```

```
DMM:CURSOR:HBARs:SElect?
```

Arguments

CURSOR1 selects the first horizontal bar cursor.

CURSOR2 selects the second horizontal bar cursor.

Examples

```
DMM:CURSOR:HBARs:SELECT CURSOR1
```

selects the first horizontal bar cursor as the active cursor.

```
DMM:CURSOR:HBARs:SELECT?
```

returns CURSOR1 when the first cursor is the active cursor.

Related Commands

CURSOR:HBARs:SElect

DMM:CURSOR:PAIred? (Query Only)

Queries and returns the data logger paired cursor measurements.

Group

Cursor

Syntax

DMM:CURSOR:PAIred?

Returns

None

Examples

DMM:CURSOR:PAIred?

might return DMM:CURSOR:PAIRED:POSITION1 0;POSITION2 0;SELECT CURSOR1.

Related Commands

CURSOR:PAIred?

DMM:CURSOR:PAIred:HDELta? (Query Only)

Queries the vertical distance between the first and second paired cursor in the data logger plot. The response is the absolute value of the difference between the vertical positions of the first and second cursors.

Group

DMM Logger

Syntax

DMM:CURSOR:PAIred:HDELta?

Returns

None

Examples

DMM:CURSOR:PAIred:HDELTA?

might return 5.08E+0 for the voltage difference between the two cursors.

Related Commands

DMM:CURSOR:FUNCTION, CURSOR:PAIred:HDELTA?

DMM:CURSor:PAIred:HPOS1? (Query Only)

Queries the vertical position of the first paired cursor in the data logger plot.

Group

DMM Logger

Syntax

DMM:CURSor:PAIred:HPOS1?

Returns

None

Examples

DMM:CURSOR:PAIRED:HPOS1?

might return $-64.0E2$, indicating that the first cursor is at -64.0 vertical units.

Related Commands

DMM:CURSor:FUNcTION, CURSor:PAIred:HPOS1?

DMM:CURSor:PAIred:HPOS2? (Query Only)

Queries the vertical position of the second paired cursor in the data logger plot.

Group

DMM Logger

Syntax

DMM:CURSor:PAIred:HPOS2?

Returns

None

Examples

DMM:CURSOR:PAIRED:HPOS2?

might return $-64.0E2$, indicating the second cursor is at -64.0 vertical units.

Related Commands

DMM:CURSOR:FUNCTION, CURSOR:PAIRED:HPOS2?

DMM:CURSOR:PAIRED:POSITION1

Sets or queries the horizontal position of the first paired cursor.

Group

DMM Logger

Syntax

DMM:CURSOR:PAIRED:POSITION1 < NR3 >

DMM:CURSOR:PAIRED:POSITION1?

Arguments

<NR3> specifies the position of the first paired cursor.

Examples

DMM:CURSOR:PAIRED:POSITION1 2.4E1

sets the first paired cursor to 24 s.

DMM:CURSOR:POSITION1?

might return $5.78E3$, indicating that the first paired cursor is at 5,780 s.

Related Commands

DMM:CURSOR:FUNCTION

DMM:CURSor:PAIred:POSITION2

Sets or queries the horizontal position of the second paired cursor.

Group

DMM Logger

Syntax

DMM:CURSor:PAIred:POSITION2 < NR3 >

DMM:CURSor:PAIred:POSITION2?

Arguments

<NR3> specifies the position of the second paired cursor.

Examples

DMM:CURSOR:PAIRED:POSITION2 2.4E1

sets the first paired cursor to 24 s.

DMM:CURSOR:POSITION2?

might return 5.78E3, indicating that the second paired cursor is at 5,780 s.

Related Commands

DMM:CURSor:FUNcTion

DMM:CURSor:PAIred:SElect

Selects the active paired cursor in the data logger plot. The active cursor appears as a solid vertical line. The unselected cursor appears as a dashed vertical line. This command is equivalent to pressing the TOGGLE button on the front panel.

Group

DMM Logger

Syntax

DMM:CURSor:PAIred:SElect { CURSOR1 | CURSOR2 }

DMM:CURSOR:PAIred:SElect?

Arguments

CURSOR1 specifies the first paired cursor.

CURSOR2 specifies the second paired cursor.

Examples

DMM:CURSOR:PAIRED:SELECT CURSOR2
selects the second paired cursor as the active cursor.

DMM:CURSOR:PAIRED:SELECT?
returns CURSOR1 when the first paired cursor is the active cursor.

Related Commands

CURSOR:PAIred:SElect

DMM:CURSOR:PAIred:VDELta? (Query Only)

Queries the vbar (time) distance between paired cursors in the data logger plot.

Group

DMM Logger

Syntax

DMM:CURSOR:PAIred:VDELta?

Returns

None

Examples

DMM:CURSOR:PAIRED:VDELTA?
might return 1.064E+00, indicating that the time between the paired cursors is 1.064 seconds.

Related Commands

DMM:CURSOR:FUNCTION, CURSOR:PAIred:VDELta?

DMM:CURSor:VBArS

Positions the data logger vertical bar cursors and CURSor:VBArS? returns the current vertical bar cursor settings for horizontal position, delta, cursor selection, and units.

Group

DMM Logger

Syntax

DMM:CURSor:VBArS SNAp

DMM:CURSor:VBArS?

Arguments

SNAp positions the vertical bar cursors at DATA:STARt and DATA:STOP.

Examples

DMM:CURSOR:VBARS SNAP

specifies that the cursors' positions are the same as the current DATA:START and DATA:STOP values.

DMM:CURSOR:VBARS?

might return :CURSOR:VBARS:UNITS SECONDS;POSITION1 1.00E1; POSITION2 9.00E1;SELECT CURSOR2.

Related Commands

DATA:STARt, DATA:STOP, CURSor:VBArS?

DMM:CURSor:VBArS:DELTA? (Query Only)

Returns the time between the two vertical bar cursors in the data logger plot.

Group

DMM Logger

Syntax

DMM:CURSor:VBARs:DELTA?

Returns

<NR3>

Examples

DMM:CURSOR:VBARS:DELTA?

might return 1.06E+02, indicating that the time between the vertical bar cursors is 106 seconds.

Related Commands

CURSor:VBARs:DELta?

DMM:CURSor:VBARs:POSITION1

Sets or queries the horizontal position of both the first vertical bar cursor and first paired cursor.

Group

DMM Logger

Syntax

DMM:CURSor:VBARs:POSITION1 <NR3>

DMM:CURSor:VBARs:POSITION1?

Arguments

<NR3> specifies the cursor position in seconds. The position is relative to the right edge of the logger plot.

Examples

DMM:CURSOR:VBARS:POSITION1 9

positions the first vertical bar or paired cursor at 9 s.

DMM:CURSOR:VBARS:POSITION1?

might return 1.27E4, indicating the first vertical bar or paired cursor is at 12,700 s.

Related Commands

CURSor:VBARs:POSITION<x>

DMM:CURSor:VBARs:POSITION2

Sets or queries the horizontal position of both the second vertical bar cursor and second paired cursor.

Group

DMM Logger

Syntax

DMM:CURSor:VBARs:POSITION2 <NR3>

DMM:CURSor:VBARs:POSITION2?

Arguments

<NR3> specifies the cursor position in seconds. The position is relative to the right edge of the logger plot.

Examples

DMM:CURSOR:VBARS:POSITION2 9

positions the second vertical bar or paired cursor at 9 s.

DMM:CURSOR:VBARS:POSITION2?

might return 1.27E4, indicating the second vertical bar or paired cursor is at 12,700 s.

Related Commands

CURSor:VBARs:POSITION<x>

DMM:CURSor:VBArS:SElect

Selects which data logger vertical bar cursor is active. The active cursor is displayed as a solid vertical line and is moved using the front-panel +/- rocker. The unselected cursor is displayed as a dashed vertical line. This command is equivalent to pressing the TOGGLE button on the front panel.

Group

DMM Logger

Syntax

```
DMM:CURSor:VBArS:SElect { CURSOR1 | CURSOR2 }
```

```
DMM:CURSor:VBArS:SElect?
```

Arguments

CURSOR1 specifies the first vertical bar cursor.

CURSOR2 specifies the second vertical bar cursor.

Examples

```
DMM:CURSOR:VBARS:SELECT CURSOR2
```

selects the second vertical bar cursor as the active cursor.

```
DMM:CURSOR:VBARS:SELECT?
```

returns CURSOR1 when the first vertical bar cursor is the active cursor.

Related Commands

CURSor:VBArS:SElect

DMM:CURVe?

Transfers logger data (waveform) from the instrument in binary or ASCII format. The waveform that is transferred has an associated waveform preamble that contains information such as data format and scale. Refer to the DMM:WFMPre command starting on page 2-141 for information about the waveform preamble. The data format is specified by the DATA:ENCdg and DATA:WIDTH commands.

The `DMM:CURVe?` query transfers data from the instrument. The data source is that which is currently displayed, either logger data or saved data. The first and last data points that are transferred are specified by the `DATA:START` and `DATA:STOP` commands.

A description of the waveform transfer process, for both oscilloscope and meter waveforms, starts on page 2–26.

Group

DMM Logger

Syntax

`DMM:CURVe?`

Examples

`DMM:CURVE?`

might return the ASCII data

```
:CURVE 13,6,3,2,-1,-9,-14,-19,-29,-35,-67,-71,-78,-62,-50,-31,-27,-7,0,12,29,39,45,43,41,47,41,38,33,26
```

Related Commands

`DATA:START`, `DATA:STOP`, `DATA:ENCdg`, `DATA:WIDTH`,
`DMM:WFMPre`

DMM:DELTA:VALue

Set and query the $\text{Rel}\Delta$ value for the DMM readings, statistics, and data logger plot. Sending this command turns $\text{Rel}\Delta$ mode on and sets the $\text{Rel}\Delta$ value. Sending this query returns the $\text{Rel}\Delta$ value.

Group

DMM Measurement

Syntax

`DMM:DELTA:VALue <NR3>`

`DMM:DELTA:VALue?`

Arguments

<NR3> is the Rel Δ value to be used as the reference level. The range is any value less than or equal to full scale.

Returns

<NR3> is the current Rel Δ value.

Examples

DMM:DELTA:VALue 24.7 sets the reference value to 24.7 V (or 24.7 Ω , if the meter function is Ω).

DMM:DELTA:VALue?

might return DMM:DELTA:VALue 24.7, the active Rel Δ value.

Related Commands

DMM:DISplay:DELTA

DMM:DISplay? (Query Only)

Returns all DMM display parameters.

Group

Miscellaneous DMM

Syntax

DMM:DISplay?

Returns

None

Examples

DMM:DISplay? might return
:DMM:DISPLAY:DELTA OFF;GRATICULE FULL;STYLE THIN

Related Commands

None

DMM:DISplay:DELTA

Set or query the Rel Δ function. When enabled, the meter reading, statistics, and data logger plot use the Rel Δ value as their reference. When disabled, they use zero as their reference.

Group

Miscellaneous DMM

Syntax

```
DMM:DISplay:DELTA { OFF | ON | <NR1> }
```

```
DMM:DISplay:DELTA?
```

Arguments

OFF or <NR1> = 0 causes the meter reading, statistics, and data logger plot to use zero as their reference.

ON or <NR1> = 1 causes the meter reading, statistics, and data logger plot to use the Rel Δ value as their reference.

Examples

```
DMM:DISplay:DELTA ON
```

displays the meter reading, statistics, and data logger plot relative to the Rel Δ value.

Related Commands

```
DMM:DELTA:VALue
```

DMM:DISplay:GRAticule

Selects the type of DMM graticule that is displayed. This command is equivalent to setting Graticule in the Display menu.

Group

Miscellaneous DMM

Syntax

```
DMM:DISplay:GRAticule { FRAMe | FULL | GRId | CROss }
```

DMM:DISplay:GRAticule?

Arguments

FRame specifies just a frame.

FULl specifies a frame, a grid, and cross hairs.

GRId specifies a frame and grid only.

CR0ss specifies a frame and cross hairs only.

Examples

DMM:DISPLAY:GRATICULE FRAME

sets the graticule type to display the frame only.

DMM:DISPLAY:GRATICULE?

returns FULL when all graticule elements (grid, frame, and cross hairs) are selected.

Related Commands

None

DMM:DISplay:STYle

Set or query the vertical thickness of the data logger plot. A Thin plot is one pixel thick. A Thick plot is three pixels thick.

Group

Miscellaneous DMM

Syntax

DMM:DISplay:STYle { THIN | THICK }

DMM:DISplay:STYle?

Arguments

THIN sets the data logger plot to one pixel thick.

THICK sets the data logger plot to three pixels thick.

Examples

DMM:DISPlay:STYle THIN

sets the data logger plot to one pixel thick.

Related Commands

None

DMM:FREQuency:REJect

Set or query the DMM frequency reject function. This function improves the repeatability of AC measurements in the presence of 50 Hz or 60 Hz noise. This command is the same as setting Noise Reject in the DMM Vertical menu.

Group

Miscellaneous DMM

Syntax

DMM:FREQuency:REJect { 50 | 60 }

DMM:FREQuency:REJect?

Arguments

50 sets the DMM to reject 50 Hz noise.

60 sets the DMM to reject 60 Hz noise.

Examples

DMM:FREQuency:REJect 60

sets the DMM to reject 60 Hz noise.

DMM:FREQuency:REJect?

might return the string DMM:FREQUENCY:REJECT 50

Related Commands

None

DMM:FUNCTION

Set or query the DMM function. The choices are DCV, ACV, resistance, continuity test, and diode test.

Group

Miscellaneous DMM

Syntax

DMM:FUNCTION { ACV | DCV | OHMs | CONTinuity | DIODE }

DMM:FUNCTION?

Arguments

ACV selects the AC volts meter function.

DCV selects the DC volts meter function.

OHMs selects the resistance meter function.

CONTinuity selects the continuity test meter function.

DIODE selects the diode (semiconductor junction) test meter function.

Examples

DMM:FUNCTION OHMs
selects the resistance meter function.

DMM:FUNCTION?
might return DMM:FUNCTION ACV if the meter function is AC volts.

Related Commands

None

DMM:HOLD

Set or query the DMM hold function, which freezes the meter reading and statistics (acquisition continues in the background). This is the equivalent of pressing the front-panel HOLD button.

Group

Miscellaneous DMM

Syntax

DMM:HOLD { OFF | ON | RUN | STOP | <NR1> }

DMM:HOLD?

Arguments

OFF or RUN or <NR1> = 0 freezes the meter reading and measurements (acquisition continues in the background).

ON or STOP or <NR1> \neq 0 resets the measurement statistics and the data logger plot and then resumes updates of the meter readings (assuming the .

Examples

DMM:HOLD OFF:

resets the measurement statistics and the data logger plot and then resumes updates of the meter readings.

DMM:HOLD?

returns 1 if the meter display is frozen.

Related Commands

ACQuire:STATE, DMM:STATE

DMM:LOGGer:MODE

This command sets and queries the DMM data logger acquisition mode.

Group

DMM Logger

Syntax

DMM:LOGGer:MODE { SAMple | PEAKdetect | AVErage }

DMM:LOGGer:MODE?

Arguments

SAMple specifies that the displayed data point value is simply the last meter reading that was taken during the acquisition interval. In sample mode, all logger data has 8 bits of precision. You can request 16 bit data with a DMM:LOGGer:CURVe? query, but the lower-order 8 bits of data will be zero. SAMple is the default mode.

PEAKdetect specifies the display of the high-low range of meter readings taken from each acquisition interval. The data logger displays the high-low range as a vertical column that extends from the highest to the lowest value sampled during the acquisition interval. PEAKdetect mode can reveal narrow spikes.

AVErage specifies averaging mode, where the resulting data logger plot shows the average of all meter readings taken during each acquisition interval.

Examples

DMM:LOGGer:MODE AVErage

sets the data logger mode to display a plot consisting of the average of meter readings from each acquisition interval.

DMM:LOGGer:MODE?

might return AVERAGE.

Related Commands

None

DMM:LOGGer:SECDiv

Set or query the data logger horizontal scale factor (s/div). The range is 24 h/div to 30 s/div (four minutes to eight days, full scale). This command is equivalent to setting SEC/DIV for the data logger.

Group

DMM Logger

Syntax

DMM:LOGGer:SECDiv <NR3>

DMM:LOGGer:SECDiv?

Arguments

<NR3> is the time per division in seconds. Valid values are listed in the table below. Values that are not in the sequence below are set to the closest valid value.

Valid Argument Values	Data Logger Time Per Division
3.0E1	30 seconds
6.0E1	1 minute
9.0E1	1.5 minutes
1.5E2	2.5 minutes
3.0E2	5 minutes
4.5E2	7.5 minutes
9.0E2	15 minutes
1.8E3	30 minutes
3.6E3	1 hour
5.4E3	1.5 hours
1.08E4	3 hours
2.16E4	6 hours
4.32E4	12 hours
8.64E4	24 hours

Examples

DMM:LOGGer:SECDiv 9E2

sets the data logger horizontal scale to 900 s (15 minutes) per division.

Related Commands

None

DMM:MEASUrement? (Query Only)

Return all DMM measurement statistic parameters.

Group

DMM Measurement

Syntax

DMM:MEASUrement?

Returns

None

Examples

DMM:MEASUREMENT?

```
might return :DMM:MEASUREMENT:MEAS1:TYPE MAXIMUM;  
UNITS "V";STATE 0;:DMM:MEASUREMENT:MEAS2:TYPE MAXIMUM;  
UNITS "V";STATE 0;:DMM:MEASUREMENT:MEAS3:TYPE MAXIMUM;  
UNITS "V";STATE 0
```

Related Commands

MEASUrement?

DMM:MEASUrement:AVERage? (Query Only)

Returns the value of the DMM Average statistic.

Group

DMM Measurement

Syntax

DMM:MEASUrement:AVERage?

Returns

<NR3>

Examples

DMM:MEASUREMENT:AVERAGE?

returns the value 2.45E1 if the average value is 24.5 V.

Related Commands

None

DMM:MEASUREMENT:CONTINUITY? (Query Only)

Return the current continuity measurement result.

Group

DMM Measurement

Syntax

DMM:MEASUREMENT:CONTINUITY?

Returns

OPEN if an open circuit is measured ($\approx >50 \Omega$).

SHORT if a closed circuit is measured ($\approx <50 \Omega$).

Examples

DMM:MEASUREMENT:CONTINUITY?

might return SHORT when circuit continuity is detected.

Related Commands

None

DMM:MEASUREMENT:DELTA? (Query Only)

Return the DMM $\Delta 0$ value, the value used for Rel Δ measurements. This query is identical to the DMM:DELTA:VALUE? query. Use the DMM:DELTA:VALUE command to set the value.

Group

DMM Measurement

Syntax

DMM:MEASUrement:DELTA?

Returns

<NR3>

Examples

DMM:MEASUrement:DELTA?
might return the value 3.94.

Related Commands

DMM:DELTA:VALue

DMM:MEASUrement:MAX? (Query Only)

Return the value of the DMM Max statistic.

Group

DMM Measurement

Syntax

DMM:MEASUrement:MAX?

Returns

<NR3>

Examples

DMM:MEASUrement:MAX?
might return the value 2.69.

Related Commands

None

DMM:MEASUrement:MEAS<x>:STATE

Set or query the state the DMM measurements (either on and off).

The source of DMM measurements (statistics) is all meter readings since the data was last reset. The data may include values no longer displayed in the data logger plot.

Group

DMM Measurement

Syntax

```
DMM:MEASUrement:MEAS<x>:STATE { OFF | ON | <NR1> }
```

```
DMM:MEASUrement:MEAS<x>:STATE?
```

Arguments

OFF or <NR1> = 0 turns measurements off.

ON or <NR1> \neq 0 turns measurements on.

Examples

```
DMM:MEASUREMENT:MEAS1:STATE ON  
turns measurement defined as MEAS1 on.
```

```
MEASUREMENT:MEAS3:STATE?  
returns either 0 or 1, indicating the state of the third measurement.
```

Related Commands

None

DMM:MEASUrement:MEAS<x>:TYPE

Sets or queries the DMM measurement type for the measurement specified by <x>. This is equivalent to selecting the measurement in the Select Statistic pop-up menu.

Group

DMM Measurement

Syntax

```
DMM:MEASUrement:MEAS<x>:TYPE { AVErage | DELTA | MAXimum  
| MINimum | PK2pk }
```

```
MEASUrement:MEAS<x>:TYPE?
```

Arguments

AVErage selects the average statistic.

DELTA selects Rel Δ statistic (value).

MAXimum selects the maximum statistic.

MINimum is minimum statistic.

PK2pk is peak-to-peak (maximum – minimum) statistic.

Examples

```
MEASUREMENT:MEAS2:TYPE MAX
```

specifies the second measurement to be the maximum statistic.

Related Commands

```
DMM:MEASUrement:MIN?, DMM:MEASUrement:MAX?,  
DMM:MEASUrement:PK2pk?, DMM:MEASUrement:AVG?,  
DMM:MEASUrement:DELTA?
```

DMM:MEASUrement:MEAS<x>:UNIts? (Query Only)

Returns the units for the DMM statistic specified by MEASUrement:MEAS<x>:TYPE.

Group

DMM Measurements

Syntax

```
DMM:MEASUrement:MEAS<x>:UNIts?
```

Returns

<QString> returns "V" for volts or "A" for amperes.

Examples

DMM:MEASUREMENT:MEAS3:UNITS?

might return "V", indicating the unit for DMM statistic 3 is volts.

Related Commands

None

DMM:MEASUREMENT:MEAS<x>:VALUE? (Query Only)

Returns the value of the DMM measurement statistic specified by <x>.

Group

DMM Measurements

Syntax

DMM:MEASUREMENT:MEAS<x>:VALUE?

Returns

<NR3>

Examples

DMM:MEASUREMENT:MEAS2:VALUE?

might return the value 5.98E2.

Related Commands

DMM:MEASUREMENT:MEAS<x>:TYPE

DMM:MEASUREMENT:MIN? (Query Only)

Return the value of the DMM Min statistic.

Group

DMM Measurement

Syntax

DMM:MEASUREMENT:MIN?

Returns

<NR3>

Examples

DMM:MEASUREMENT:MIN?
might return the value -8.69.

Related Commands

None

DMM:MEASUREMENT:PK2pk? (Query Only)

Return the value of the DMM Max–Min statistic.

Group

DMM Measurement

Syntax

DMM:MEASUREMENT:PK2pk?

Returns

<NR3>

Examples

DMM:MEASUREMENT:PK2pk?
might return the value 5.98.

Related Commands

None

DMM:PROBE? (Query Only)

Returns all DMM probe parameters.

Group

DMM Probe

Syntax

DMM:PROBe?

Returns

None

Examples

DMM:PROBe? might return
:DMM:PROBE:UNITS VOLTS;CURRENTSCALE 1.0E-3

DMM:PROBe:CURrentscale

Set or query the current probe conversion factor.

Group

DMM Probe

Syntax

DMM:PROBe:CURrentscale <NR3>

DMM:PROBe:CURrentscale?

Arguments

<NR3> is a value listed in the table below. Values other than those listed are set to the closest valid value.

<NR3>	Conversion Factor (Transimpedance)
1.0E-3	1 mV/A
1.0E-2	10 mV/A
1.0E-1	100 mV/A
1.0E0	1 V/A

Examples

DMM:PROBe:CURrentscale 1.0E-2
sets the scale for a 10 mV/A current probe.

Related Commands

DMM:PROBe:UNITs

DMM:PROBe:UNITs

Set or query the DMM probe type to be voltage or current.

Group

DMM Probe

Syntax

DMM:PROBe:UNITs { VOLts | AMPS }

DMM:PROBe:UNITs?

Arguments

VOLts sets the DMM probe to a voltage probe (meter leads).

AMPS sets the DMM probe to a current probe .

Examples

DMM:PROBe:UNITs AMPS

sets the DMM probe to be a current probe.

Related Commands

DMM:PROBe:CURrentscale

DMM:RANGE:ACV

Set or query the DMM AC voltage range.

Group

DMM Function Range

Syntax

DMM:RANGE:ACV <NR3>

DMM:RANGE:ACV?

Arguments

<NR3> is a one of the following values: 0.4, 4, 40, 400, or 640. Values other than those listed are set to the closest valid value.

Examples

DMM:RANGE:ACV 40 sets the full scale range to 40 V.

Related Commands

None

DMM:RANGE:AUTO

Set or query the DMM autorange function on or off.

Group

DMM Function Range

Syntax

DMM:RANGE:AUTO { OFF | ON }

DMM:RANGE:AUTO?

Arguments

OFF turns the autorange function off.

ON turns the autorange function on.

Examples

DMM:RANGE:AUTO?
might return ON if the DMM autorange function is on.

Related Commands

None

DMM:RANGE:DCV

Set or query the DMM DC voltage range.

Group

DMM Function Range

Syntax

DMM:RANGE:DCV <NR3>

DMM:RANGE:DCV?

Arguments

<NR3> is a one of the following numbers: 0.4, 4, 40, 400, or 880. Values other than those listed are set to the closest valid value.

Examples

DMM:RANGE:DCV 4E-1 sets the full scale range to ± 400 mV.

Related Commands

None

DMM:RANGE:OHMS

Set or query the DMM resistance range.

Group

DMM Function Range

Syntax

DMM:RANGE:OHMS <NR3>

DMM:RANGE:OHMS?

Arguments

<NR3> is a one of the following values: 4.0E2, 4.0E3, 4.0E4, 4.0E5, 4.0E6, or 4.0E7. Values other than those listed are set to the closest valid value.

Examples

DMM:RANGE:OHMS 4E4 sets the full scale range to 40 k Ω .

Related Commands

None

DMM:RECALL:DATA

Recalls the logger data from a stored location to the display. Query form returns stored data location (source) if data is recalled.

Group

DMM Data

Syntax

DMM:RECALL:DATA ST0[1-10]

DMM:RECALL:DATA?

Arguments

ST0[1-10] specifies a stored location in the range from 1 through 10.

Returns

0 if no data is recalled.

[1-10] indicates the stored data location (source) if data is recalled.

Examples

DMM:RECALL:DATA 4
recalls stored location 4 to the meter display.

Related Commands

DMM:SAVE:DATA

DMM:RESet (No Query Form)

Reset the DMM data logger and measurement statistics.

Group

Miscellaneous DMM

Syntax

DMM:RESet

Arguments

None

Examples

None

Related Commands

None

DMM:SAVe:DATa (No Query Form)

Saves all DMM data to a stored location. DMM data storage locations are separate from the oscilloscope waveform storage locations.

Group

Syntax

DMM:SAVe:DATa ST0[1-10]

Arguments

ST0[1-10] specifies a stored location in the range from 1 through 10.

Examples

DMM:SAVe:DATa ST02

saves the data logger plot, current measurement, and statistics to stored location 2.

Related Commands

DMM:RECAll:DATa

DMM:STATE? (Query Only)

Returns the status of the DMM display.

Group

Miscellaneous DMM

Syntax

DMM:STATE?

Returns

RUN indicates that the meter is updating its reading, measurement statistics, and data logger plot, and that the range is fixed.

HOLD indicates the meter reading, measurement statistics, and data logger plot are frozen.

AUTO indicates that the meter is updating its reading, statistics, and data logger plot, and that the autorange function is on.

SAVED indicates the displayed meter reading, measurement statistics, and data logger plot are recalled from memory.

Examples

DMM:STATE?

might return AUTO if the autorange function is on.

Related Commands

DMM:FUNcTion

DMM:WAVFrm? (Query Only)

Returns DMM:WFMPre? and DMM:CURVe? data specified by the DMM:DATA:SOUrce command. This command is equivalent to sending DMM:WFMPre?; DMM:CURVe?.

Group

DMM Logger

Syntax

DMM:WAVFrm?

Returns

See DMM:WFMPre? and DMM:CURVe? commands.

Examples

None

Related Commands

DMM:CURVe?, DMM:WFMPre?

DMM:WFMPre? (Query Only)

Returns the waveform formatting data for the DMM waveform currently displayed.

Group

DMM Logger

Syntax

DMM:WFMPre?

Returns

The format of the response is:

```
:DMM:WFMPRE:WFID "Logger, VDC, 10.000000 V/div, .500000  
min/div, 480 points, Rel Delta=.000000, Peak  
Detect";NR_PT 2500;PT_FMT ENVELOPE;XUNIT "s";  
XINCR 5.0E-1;PT_OFF 0;YUNIT "V";YMULT 5.0E-1;  
YOFF 0.0E0;YZERO 0.0E0
```

Examples

None

Related Commands

DMM:WAVFrm?

DMM:WFMPre:WFId? (Query Only)

Returns information about the waveform such as source, function, vertical units/division, time/division, delta value, and acquisition mode.

Group

DMM Logger

Syntax

DMM:WFMPre:WFId?

Returns

None

Examples

```
DMM:WFMPre:WFId?  
might return :DMM:WFMPRE:LOGGER:WFID "Logger, VDC,  
10.000000 V/div, .500000 min/div, 480 points,  
Rel Delta=.000000, Peak Detect"
```

Related Commands

DMM:WFMPre

DMM:VALue? (Query Only)

Returns the current meter reading.

Group

DMM Measurement

Syntax

DMM:VALue?

Returns

<NR3> is the current meter reading

Examples

DMM:VALue?

returns -2.534E1 if the current meter reading is -25.34 VDC.

Related Commands

None

*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 more detailed discussion of the use of these registers, see page 3–1.

Group

Status and Error

Syntax

*ESE <NR1>

*ESE?

Arguments

<NR1> is a value in the range from 0 through 255. The binary bits of the ESER are set according to this value.

The power-on default for ESER is 0 if *PSC is 1. If *PSC is 0, the ESER maintains its value through a power cycle.

NOTE. *Setting the DESER and the 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 DESE command to set the DESER. A discussion of event handling begins on page 3–1.*

Examples

*ESE 209

sets the ESER to binary 11010001, which enables the PON, URQ, EXE, and OPC bits.

*ESE?

might return the string *ESE 186, showing that the ESER contains the binary value 10111010.

Related Commands

*CLS, DESE, *ESR?, EVENT?, EVMsg? *SRE, *STB?

***ESR? (Query Only)**

Returns the contents of the Standard Event Status Register (SESR). *ESR? also clears the SESR (since reading the SESR clears it). For a more detailed discussion of the use of these registers see page 3–1.

Group

Status and Error

Syntax

*ESR?

Returns

None

Examples

*ESR?

might return the value 213, showing that the SESR contains binary 11010101.

Related Commands

ALLEv?, *CLS, DESE, *ESE, EVENT?, EVMsg?, *SRE, *STB?

EVENT? (Query Only)

Returns from the Event Queue an event code that provides information about the results of the last *ESR? read. EVENT? also removes the returned value from the Event Queue. A discussion of event handling begins on page 3–1.

Group

Status and Error

Syntax

EVENT?

Returns

None

Examples

EVENT?

might return the response :EVENT 110, showing that there was an error in a command header.

Related Commands

ALLEV?, *CLS, DESE, *ESE, *ESR?, EVMsg?, *SRE, *STB?

EVMsg? (Query Only)

Removes from the Event Queue a single event code associated with the results of the last *ESR? read, and returns the event code along with an explanatory message. A more detailed discussion of event handling begins on page 3–1.

Group

Status and Error

Syntax

EVMsg?

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 instrument. As much of the command as possible is returned without exceeding the 60 character limit of the <Message> and <Command> strings combined. The command string is right-justified.

Examples

EVMSG?

might return the message :EVMSG 110,"Command header error".

Related Commands

ALLEv?, *CLS, DESE, *ESE, *ESR?, EVENT?, *SRE, *STB?

EVQty? (Query Only)

Returns the number of event codes that are in the Event Queue. This is useful when using ALLEv? since it lets you know exactly how many events will be returned.

Group

Status and Error

Syntax

EVQty?

Returns

<NR1>

Examples

EVQTY?

might return 3 as the number of event codes in the Event Queue.

Related Commands

ALLEv?, EVENT?, EVMsg?

FACTory (No Query Form)

Resets the instrument to its factory default settings. This command is identical to the *RST command.

Group

Miscellaneous Oscilloscope

Syntax

FACTory

Setting the instrument to factory default has the following impact on the programming interface:

- Clears the Event Status Enable Register
- Clears the Service Request Enable Register
- Sets the Device Event Status Enable Register to 255
- Sets the Power On Status Clear Flag to TRUE
- Purges all defined aliases
- Enables all Command Headers (HEADer ON)
- Sets the macro defined by *DDT to a “zero-length field”
- Clears the pending operation flag and associated operations

The FACTory command does not alter the following:

- The state of the RS-232 interface
- Calibration data that affects device specifications
- Protected user data
- Stored settings
- Stored waveforms or data
- The current password (if implemented)
- Hard copy parameters

Arguments

None

Examples

None

Related Commands

*PSC, *RCL, RECALL:SETUp, *RST, *SAV, SAVe:SETUp

HARDCopy

Sends a copy of the screen display followed by an EOI to the port specified by HARDCopy:PORT. The format and layout of the output is specified with the HARDCopy:FORMAt and HARDCopy:LAYout commands. This command is equivalent to pressing the front-panel HARD COPY button.

HARDCopy? returns format, layout, and port information.

NOTE. *This command is not IEEE Std 488.2–1987 compatible.*

Group

Hard copy

Syntax

HARDCopy { ABOrt | CLEARSpool | START }

HARDCopy?

Arguments

ABOrt terminates the hard copy output in process.

NOTE. *<BREAK> does not clear the output queue once a hard copy is in process. The only way to abort the hard copy process is to send the HARDCopy ABOrt command. The output queue can then be cleared using <BREAK>.*

CLEARSpool clears the printer output spooler.

START initiates a screen copy that is sent to the controller where it can be stored in a file or redirected to a printing device.

NOTE. Use the *WAI command between *HARDCopy START* commands to ensure that the first hard copy is complete before starting another.

Examples

HARDCOPY ABORT

stops any hard copy output that is in process.

Related Commands

None

HARDCopy:FORMat

Selects the output data format for hard copies. This command is equivalent to setting Format in the Hard Copy System branch of the Utility menu.

Group

Hard copy

Syntax

```
HARDCopy:FORMat { BMP | DESKJet | DPU411 | DPU412 |  
EPSImage | EPSOn | INTERLeaf | LASERJet | PCX | THInkjet  
| TIFF }
```

HARDCopy:FORMat?

Arguments

BMP sets the format to Microsoft Windows file format.

DESKjet sets the format to high-resolution printer format.

DPU411 sets the format to thermal printer format for DPU 411/II or HC 411.

DPU412 sets the format to thermal printer format for DPU 412.

EPSImage sets the format to encapsulated postscript image file format.

EPSOn sets the format to 9-pin or 24-pin dot matrix printer format.
INTERLeaf sets the format to Interleaf image object file format.
LASERJet sets the format to laser printer format.
PCX sets the format to PC Paintbrush monochrome image file format.
THInkjet sets the format to inkjet printer format.
TIFF sets the format to tag image file format.

Examples

HARDCOPY:FORMAT TIFF
sets the hard copy output format to TIFF.

HARDCOPY:FORMAT?
might return INTERLEAF as the hard copy output format.

Related Commands

None

HARDCopy:LAYout

Selects the printing orientation. This command is equivalent to setting Layout in the Hard Copy System branch of the Utility menu.

Group

Hard copy

Syntax

HARDCopy:LAYout { LANDscape | PORTRait }

HARDCopy:LAYout?

Arguments

LANDscape specifies that the bottom of the hard copy is along the long side of the page.

PORTRait specifies that the bottom of the hard copy is along the short side of the page. This is the standard format.

Examples

HARDCOPY:LAYOUT?

might return PORTRAIT as the page layout format of the hard copy output.

Related Commands

None

HARDCopy:PORT

Selects the output port for the printer; the RS-232 port is the only valid choice for the instrument. This command is included for compatibility with older Tektronix instruments.

Group

Hard copy

Syntax

HARDCopy:PORT RS232

HARDCopy:PORT?

Arguments

RS232 specifies that the hard copy is sent out the RS232 port. This is the only valid argument for this command.

Examples

HARDCOPY:PORT?

always returns RS232 as the selected hardcopy output port.

Related Commands

HARDCopy

HDR

This command is identical to the HEADer query and is included for compatibility with older Tektronix instruments.

Group

Miscellaneous Oscilloscope

Syntax

```
HDR { <NR1> | OFF | ON }
```

HDR?

Arguments

None

Examples

None

Related Commands

HEADer

HEADer

Sets and queries the Response Header Enable State that causes the instrument to either include or omit headers on query responses. This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk); they never return headers.

Group

Miscellaneous Oscilloscope

Syntax

```
HEADer { <NR1> | OFF | ON }
```

HEADer?

Arguments

ON or <NR1> \neq 0 sets the Response Header Enable State to true. This causes the instrument to include headers on applicable query responses. You can then use the query response as a command.

OFF or <NR1> = 0 sets the Response Header Enable State to false. This causes the instrument to omit headers on query responses so that only the argument is returned.

Examples

HEADER OFF

causes the instrument to omit headers from query responses.

HEADER?

might return the value 1, showing that the Response Header Enable State is true.

Related Commands

VERBose

HORizontal? (Query Only)

Returns all settings for the oscilloscope horizontal commands. The commands HORizontal:MAIn:SCAle, HORizontal:MAIn:SECdiv, HORizontal:SCAle, and HORizontal:SECdiv are equivalent so HORizontal:MAIn:SCAle is the only value that is returned.

Group

Horizontal

Syntax

HORizontal?

Returns

None

Examples

HORIZONTAL?

might return the string :HORIZONTAL:MODE MAIN;RECORDLENGTH 2500; POSITION 5.0E+0;TRIGGER:POSITION 50;:HORIZON-


```
TAL:MAIN:SCALE 1.0E-6;:HORIZONTAL:DELAY:MODE RUNSAF-  
TER;SCALE 1.0E-6;TIME: 16.0E-9;:HORIZONTAL:REFA  
LOCK;REFB LOCK
```

Related Commands

None

HORizontal:DELay? (Query Only)

Returns all oscilloscope horizontal delayed time base parameters. The commands `HORizontal:DELay:SECdiv` and `HORizontal:DELay:SCALE` are identical so only `HORizontal:DELay:SCALE` is returned.

Group

Horizontal

Syntax

```
HORizontal:DELay?
```

Returns

None

Examples

```
HORIZONTAL:DELAY?  
might return the delay parameters :HORIZONTAL:DELAY:MODE  
RUNSAFTER;SCALE 1.0E-6;TIME: 16.0E-9
```

Related Commands

`HORizontal?`, `HORizontal:DELay:MODE?`, `HORizontal:DELay:SCALE?`, `HORizontal:DELay:SECdiv?`, `HORizontal:DELay:TIME?`

HORizontal:DELay:MODE

Included for compatibility purposes only.

Group

Horizontal

Syntax

HORizontal:DELay:MODE RUNSAfter

HORizontal:DELay:MODE?

Arguments

RUNSAfter specifies that the delayed time base runs a user-specified amount of delay time after the main trigger event.

Examples

HORizontal:DELay:MODE?

always returns RUNSAfter, the only valid argument.

Related Commands

None

HORizontal:DELay:SCALE

Sets the time per division for the oscilloscope delayed time base. This command is equivalent to setting SEC/DIV for the delayed time base.

Group

Horizontal

Syntax

HORizontal:DELay:SCALE <NR3>

HORizontal:DELay:SCALE?

Arguments

<NR3> is the time per division. The range is 10 ns (THS 710) or 5 ns (THS 720) to 5 s in a 1–2–5 sequence. Values that are not in a 1–2–5 sequence are set to the closest valid value. If the delayed time base scale is set slower than the main time base scale, both the main and delayed time base scales are set to the delay scale value.

Examples

`HORIZONTAL:DELAY:SCALE 2.0E-6`
sets the delay scale to 2 μ s per division.

`HORIZONTAL:DELAY:SCALE 9.0E-6`
sets the delay scale to 10 μ s per division. Since 9 μ s is not a valid value within the 1–2–5 sequence, it is automatically set to the closest valid value.

`HORIZONTAL:DELAY:SCALE?`
might return 1.0E-3, indicating that the delay time is 1 ms per division.

Related Commands

`HORizontal:DELay:SECdiv`

HORizontal:DELay:SECdiv

This command is identical to the `HORizontal:DELay:SCALE` command. It is provided to maintain program compatibility with some older models of Tektronix instruments.

Group

Horizontal

Syntax

`HORizontal:DELay:SECdiv <NR3>`

`HORizontal:DELay:SECdiv?`

Arguments

None

Examples

None

Related Commands

HORizontal:DELay:SCAlE

HORizontal:DELay:TIME

Sets the delay time to wait after the oscilloscope main trigger before the delayed time base begins. This command is equivalent to setting Delayed Runs After Main in the Horizontal Time Base pop-up menu.

The query form returns the oscilloscope delay time parameter.

Group

Horizontal

Syntax

HORizontal:DELay:TIME <NR3>

HORizontal:DELay:TIME?

Arguments

<NR3> is the time, in seconds, between the main trigger and the delayed trigger. The range is from one acquired sample interval to 50 s. Resolution depends on the delayed time base setting.

Examples

HORIZONTAL:DELAY:TIME 2.0E-3

sets the delay time between the main and delayed time base to 2 ms.

HORIZONTAL:DELAY:TIME?

might return :HORIZONTAL:DELAY:TIME:16.0E-9 for the delay time.

Related Commands

HORizontal:DELay:MODE, HORizontal:DELay:TIME:RUNSAfter?

HORizontal:DELay:TIME:RUNSAfter

Sets or queries the delay time to wait after the oscilloscope main trigger before the delayed time base begins. This is equivalent to setting Delayed Runs After Main in the Horizontal Time Base pop-up menu. This command is the same as the HORizontal:DELay:TIME command.

Group

Horizontal

Syntax

```
HORizontal:DELay:TIME:RUNSAfter <NR3>
```

```
HORizontal:DELay:TIME:RUNSAfter?
```

Arguments

<NR3>, see HORizontal:DELay:TIME.

Examples

```
HORIZONTAL:DELAY:TIME:RUNSAFTER 2.0E-3
```

sets the delay time between the main and delayed time base to 2 ms.

Related Commands

```
HORizontal:DELay:MODE
```

HORizontal:FITtoscreen

The HORizontal:FITtoscreen command turns the 10X horizontal magnification on and off.

Group

Horizontal

Syntax

```
HORizontal:FITtoscreen { OFF | ON }
```

```
HORizontal:FITtoscreen?
```

Arguments

OFF displays 250 points across the screen (10X magnification is on).

ON displays all 2500 points in compressed form across the screen (10X magnification is off).

Examples

HORIZONTAL:FITtoscreen OFF
turns on the horizontal 10X magnification.

Related Commands

None

HORizontal:MAIn? (Query Only)

Returns the time per division of the oscilloscope main time base. The commands HORizontal:MAIn:SECdiv and HORizontal:MAIn:SCALE are identical so only HORizontal:MAIn:SCALE is returned.

Group

Horizontal

Syntax

HORizontal:MAIn?

Returns

None

Examples

HORIZONTAL:MAIN?
might return :HORIZONTAL:MAIN:SCALE 1.0E-6.

Related Commands

HORizontal:SCALE, HORizontal:SECdiv, HORizontal:MAIn:SECdiv, DMM:LOGGer:TPDIV

HORizontal:MAIn:SCAle

Sets the time per division for the oscilloscope main time base. This command is equivalent to setting SEC/DIV for the main time base.

Group

Horizontal

Syntax

`HORizontal:MAIn:SCAle <NR3>`

`HORizontal:MAIn:SCAle?`

Arguments

<NR3> is the time per division. The range is 10 ns (THS 710) or 5 ns (THS 720) to 5 s in a 1–2–5 sequence. Values that are not in a 1–2–5 sequence are set to the closest valid value.

Examples

`HORIZONTAL:MAIN:SCALE 2E-6`
sets the main scale to 2 μ s per division.

Related Commands

`HORizontal:DELay:SCAle`, `HORizontal:DELay:SECdiv`, `HORizontal:MAIn:SECdiv`, `DMM:LOGGer:SECDiv`

HORizontal:MAIn:SECdiv

Sets the time per division for the oscilloscope main time base. This command is identical to the `HORizontal:MAIn:SCAle` command. It is provided to maintain program compatibility with some older models of Tektronix instruments.

Group

Horizontal

Syntax

`HORizontal:MAIn:SECdiv <NR3>`

`HORizontal:MAIn:SECdiv?`

Arguments

<NR3> is the time per division. The range is 10 ns (THS 710) or 5 ns (THS 720) to 5 s in a 1–2–5 sequence. Values that are not in a 1–2–5 sequence are set to the closest valid value.

Examples

`HORIZONTAL:MAIN:SECdiv 2E-6`
sets the main scale to 2 μ s per division.

Related Commands

`HORizontal:DELay:SCAle`, `HORizontal:DELay:SECdiv`, `HORizontal:MAIn:SCAle`

HORizontal:MODE

Selects whether the oscilloscope horizontal display uses the main or delayed time base. This command is equivalent to setting Time Base in the Horizontal menu.

Group

Horizontal

Syntax

`HORizontal:MODE { DELAYEd | MAIn }`

`HORizontal:MODE?`

Arguments

`DELAYEd` means that the selected waveform is horizontally scaled relative to the delayed time base.

`MAIn` means that the waveform is horizontally scaled relative to the main time base.

Examples

`HORIZONTAL:MODE DELAYED`
uses the delayed horizontal scale to display the waveform.

Related Commands

None

HORizontal:POSition

Positions the oscilloscope waveform horizontally on the display. This is used for both main and delayed time bases. This command is equivalent to adjusting the front-panel HORIZONTAL POSITION rocker.

Group

Horizontal

Syntax

HORizontal:POSition <NR3>

HORizontal:POSition?

Arguments

<NR3> is from 0 to 99.96 and is the percent of the waveform that is displayed left of the center graticule.

Examples

HORizontal:POSITION 10

sets the horizontal position of the waveform such that 10% of the waveform is to the left of screen center.

Related Commands

None

HORizontal:REF<x>

Sets or queries the oscilloscope reference waveform position lock.

Group

Horizontal

Syntax

HORizontal:REF<x> { LOCK | INDEpendent }

`HORizontal:REF<x>?`

Arguments

`LOCK` locks the horizontal position of the reference waveform to the active waveforms.

`INDEpendent` unlocks the horizontal position of the reference waveform and allows it to be positioned independently.

Examples

`HORIZONTAL:REFA LOCK`

locks the horizontal position of Ref A to the active waveforms.

Related Commands

None

HORizontal:SCALE

Sets the time per division for the oscilloscope main time base and is identical to the `HORizontal:MAIn:SCALE` command. It is included for compatibility purposes.

Group

Horizontal

Syntax

`HORizontal:SCALE <NR3>`

`HORizontal:SCALE?`

Arguments

`<NR3>` is the time per division. The range is 10 ns (THS 710) or 5 ns (THS 720) to 5 s in a 1–2–5 sequence. Values that are not in a 1–2–5 sequence are set to the closest valid value.

Examples

`HORIZONTAL:SCALE 2E-6`

sets the main scale to 2 μ s per division.

Related Commands

DMM:LOGGer:TPDIV

HORizontal:SECdiv

Sets the time per division for the oscilloscope main time base and is identical to the HORizontal:MAIn:SCAle command. It is included for compatibility purposes.

Group

Horizontal

Syntax

HORizontal:SECdiv <NR3>

HORizontal:SECdiv?

Arguments

<NR3> is the time per division. The range is 10 ns (THS 710) or 5 ns (THS 720) to 5 s in a 1–2–5 sequence. Values that are not in a 1–2–5 sequence are set to the closest valid value.

Examples

HORIZONTAL:SECdiv 2E-6

sets the main scale to 2 μ s per division.**Related Commands**

DMM:LOGGer:TPDIV

HORizontal:TRIGger? (Query Only)

Returns the oscilloscope horizontal trigger position.

Group

Horizontal

Syntax

HORizontal:TRIGger?

Returns

None

Examples

HORIZONTAL:TRIGGER?

might return :HORIZONTAL:TRIGGER:POSITION 50.

Related Commands

HORizontal:TRIGger:POSition?

HORizontal:TRIGger:POSition

Sets or queries the position of the oscilloscope trigger. This is equivalent to setting Trigger Position in the Horizontal menu.

Group

Horizontal

Syntax

HORizontal:TRIGger:POSition <NR1>

HORizontal:TRIGger:POSition?

Arguments

<NR1> is from 0 to 100%, and is the amount of pretrigger information in the waveform.

Examples

HORIZONTAL:TRIGGER:POSITION?

might return 50.

Related Commands

HORizontal:TRIGger?

ID? (Query Only)

Returns identifying information about the instrument and its firmware in Tektronix Codes and Formats notation.

Group

Miscellaneous Oscilloscope

Syntax

ID?

Returns

The instrument identification in the following format:

TEK/<model number>,CF:91.1CT,FV:<firmware version number>

Examples

ID?

might return TEK/THS720,CF:91.1CT,FV:1.0

Related Commands

*IDN?

*IDN? (Query Only)

Returns the instrument identification code in IEEE 488.2 notation.

Group

Miscellaneous Oscilloscope

Syntax

*IDN?

Returns

The instrument identification in the following format:

TEKTRONIX,<model number>,0,CF:91.1CT FV:<firmware version number>

Examples

*IDN?

might return the response

TEKTRONIX,THS720,0,CF:91.1CT FV:1.0

Related Commands

ID?

LOCK

Enables and disables all front-panel buttons and knobs. There is no front-panel equivalent.

Group

Miscellaneous Oscilloscope

Syntax

```
LOCK { ALL | NONE }
```

LOCK?

Arguments

ALL disables all front-panel controls except the ON/STBY button.

NONE enables all front-panel controls. This is equivalent to the UNLOCK ALL command.

Examples

```
LOCK ALL
```

locks the front-panel controls.

```
LOCK?
```

returns NONE when the front-panel controls are enabled by this command.

Related Commands

UNLOCK

*LRN? (Query Only)

Returns a string listing the instrument settings, except for configuration information for the calibration values. You can use this string to return the instrument to the state it was in when you sent *LRN?.

Group

Miscellaneous Oscilloscope

Syntax

*LRN?

NOTE. *LRN? always returns a string including command headers, regardless of the setting of the HEADer command. This is because the returned string is intended to be sent back to the instrument as a command string. The VERBose command can still be used normally to specify whether the returned headers should be abbreviated.

Returns

None

Examples

*LRN?

a partial response might look like this:

```
:ACQUIRE:STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;  
STOPAFTER RUNSTOP;COUNT 1;:HEADER 1;:VERBOSE 1;  
:CURSOR:FUNCTION OFF;VBARS:UNITS SECONDS;POSITION1  
1.00E-6;POSITION2 9.00E-6;SELECT CURSOR1
```

Related Commands

HEADer, SET?, VERBose

MATH1? (Query Only)

Returns the definition for the oscilloscope math waveform.

Group

Vertical

Syntax

MATH1?

Returns

None

Examples

MATH1?

returns CH1-CH2 if the math waveform is defined as channel 1 – channel 2.

Related Commands

MATH1:DEFINE

MATH1:DEFINE

Allows the user to define a new oscilloscope waveform using a mathematical expression.

Group

Vertical

Syntax

MATH1:DEFINE <QString>

MATH1:DEFINE?

Arguments

<QString> contains the mathematical expression. The expression can include any amount of white space.

The format for a dual waveform expression is:

<source><operator><source>

where:

<operator> ::= { + | - | * }

<source> ::= CH<x>

Examples

MATH1:DEFINE "Ch1 + cH2"

sets the math waveform so that it displays the sum of channel 1 and channel 2.

Related Commands

MATH1?

MEASUrement? (Query Only)

Returns all oscilloscope measurement parameters.

Group

Oscilloscope Measurement

Syntax

MEASUrement?

Returns

None

Examples

MEASUREMENT?

```
might return :MEASUREMENT:MEAS1:STATE 0;TYPE PERIOD;
UNITS "s";SOURCE1 CH1;:MEASUREMENT:MEAS2:STATE 0;
TYPE PERIOD;UNITS "s";SOURCE1 CH1;:MEASUREMENT
:MEAS3:STATE 0;TYPE PERIOD;UNITS "s";SOURCE1 CH1;
:MEASUREMENT:MEAS4:STATE 0;TYPE PERIOD;UNITS "s";SOURCE1
CH1;:MEASUREMENT:IMMED:TYPE PERIOD;UNITS "s";SOURCE1
CH1;:MEASUREMENT:METHOD HISTOGRAM;REFLEVEL:METHOD
PERCENT;ABSOLUTE:HIGH 0.0E+0;LOW 0.0E+0;MID 0.0E+0;
:MEASUREMENT:REFLEVEL:PERCENT:HIGH 90.0E+0;LOW 10.0E+0;
MID 50.0E+0
```

Related Commands

DMM:MEASUrement?

MEASUrement:GATing

Sets or queries measurement gating. Use the horizontal bar cursors to define the measurement start and stop locations. This function is not available through the front-panel interface.

Group

Oscilloscope Measurement

Syntax

```
MEASUrement:GATing { OFF | ON | <NR1> }
```

```
MEASUrement:GATing?
```

Arguments

ON or <NR1> = 1 turns on measurement gating.

OFF or <NR1> = 0 turns off measurement gating.

Examples

```
MEASUrement:GATing?
```

returns MEASUrement:GATING 1 if measurement gating is on.

Related Commands

None

MEASUrement:IMMed? (Query Only)

Returns all immediate oscilloscope measurement setup parameters.

Group

Oscilloscope Measurement

Syntax

```
MEASUrement:IMMed?
```

Returns

None

Examples

```
MEASUREMENT:IMMED?
```

might return :MEASUREMENT:IMMED:TYPE PERIOD;UNITS "s";
SOURCE1 CH1

Related Commands

None

MEASUREMENT:IMMED:SOURCE[1]

Sets or queries the source for all immediate oscilloscope measurements.

Group

Oscilloscope Measurement

Syntax

```
MEASUREMENT:IMMED:SOURCE[1] { CH<x> | MATH1 | REF<x> }
```

```
MEASUREMENT:IMMED:SOURCE[1]?
```

Arguments

CH<x> is an input channel.

MATH1 is the math waveform.

REF<x> is a reference waveform.

Examples

```
MEASUREMENT:IMMED:SOURCE MATH1  
specifies MATH1 as the immediate measurement source.
```

Related Commands

None

MEASUREMENT:IMMED:TYPE

Specifies the immediate oscilloscope measurement.

Group

Oscilloscope Measurement

Syntax

```
MEASUREMENT:IMMED:TYPE { AMPLitude | BURst | CMEan |  
CRMs | FALL | FREQuency | HIGH | LOW | MAXimum | MEAN |  
MINimum | NDUTy | NOVershoot | NWidth | PDUTy | PERIOD |  
PK2pk | POVershoot | PWidth | RISE | RMS }
```

MEASUREMENT:IMMED:TYPE?

Arguments

AMPLITUDE is the high value minus the low value.

BURST is the time from the first MidRef crossing to the last MidRef crossing.

CMEAN is the arithmetic mean over one cycle.

CRMS is the true Root Mean Square value over one waveform cycle.

FALL is the time that it takes for the falling edge of a pulse to fall from a HighRef value to a LowRef value.

FREQUENCY is the reciprocal of the period measured in Hertz.

HIGH is the 100% reference level.

LOW is the 0% reference level.

MAXIMUM is the highest amplitude.

MEAN is the arithmetic mean over the entire waveform.

MINIMUM is the lowest amplitude.

NDUTY is the ratio of the negative pulse width to the signal period expressed as a percentage.

NOVERSHOOT is the negative overshoot, expressed as:

$$NOVERSHOOT = 100 \times \frac{(Low - Minimum)}{Amplitude}$$

NWIDTH is the distance (time) between MidRef (usually 50%) amplitude points of a negative pulse.

PDUTY is the ratio of the positive pulse width to the signal period expressed as a percentage.

PERIOD is the time, in seconds, it takes for one complete signal cycle to happen.

PK2PK is the absolute difference between the maximum and minimum amplitude.

POVershoot is the positive overshoot, expressed as:

$$POVershoot = 100 \times \frac{(Maximum - High)}{Amplitude}$$

PWIdth is the distance (time) between MidRef (usually 50%) amplitude points of a positive pulse.

RISe is the time that it takes for the leading edge of a pulse to rise from a low reference value to a high reference value.

RMS is the true Root Mean Square value.

Examples

```
MEASUREMENT:IMMED:TYPE FREQUENCY
```

defines the immediate measurement to be a frequency measurement.

Related Commands

None

MEASUREMENT:IMMED:UNITS? (Query Only)

Returns the units for the immediate oscilloscope measurement.

Group

Oscilloscope Measurement

Syntax

```
MEASUREMENT:IMMED:UNITS?
```

Returns

<QString> returns "V" for volts, "A" for amperes, "VV" for volts squared, "AA" for amperes squared, "W" for watts, "s" for seconds, "Hz" for hertz, or "%" for percent.

Examples

```
MEASUREMENT:IMMED:UNITS?
```

might return "s", indicating that the units for the immediate measurement are seconds.

Related Commands

MEASUrement:IMMed:TYPe

MEASUrement:IMMed:VALue? (Query Only)

Executes the immediate oscilloscope measurement specified by the MEASUrement:IMMed:TYPe command. The measurement is taken on the source specified by the MEASUrement:IMMed:SOUrce command.

Group

Oscilloscope Measurement

Syntax

MEASUrement:IMMed:VALue?

Returns

<NR3>

Examples

MEASUrement:IMMed:VALue?

might return 28.75E6 if you are measuring the frequency of a 28.76 MHz signal.

Related Commands

None

MEASUrement:MEAS<x>? (Query Only)

Returns all measurement parameters for the displayed oscilloscope measurement specified by <x>, where <x> is 1 through 4.

Group

Oscilloscope Measurement

Syntax

MEASUrement:MEAS<x>?

Returns

None

Examples

```
MEASUREMENT:MEAS3?
```

```
might return :MEASUREMENT:MEAS3:STATE 0;TYPE PERIOD;  
UNITS "s";SOURCE1 CH1.
```

Related Commands

None

MEASUREMENT:MEAS<x>:SOURCE[1]

Sets or queries the source for all single channel oscilloscope measurements.

Group

Oscilloscope Measurement

Syntax

```
MEASUREMENT:MEAS<x>:SOURCE[1] { CH<x> | MATH1 |  
REF<x> }
```

```
MEASUREMENT:MEAS<x>:SOURCE[1]?
```

Arguments

CH<x> is an input channel.

MATH1 is the math waveform.

REF<x> is a reference waveform.

Examples

```
MEASUREMENT:MEAS2:SOURCE1 MATH1
```

specifies MATH1 as the measurement 2 source.

Related Commands

None

MEASUREMENT:MEAS<x>:STATE

Controls the oscilloscope measurement system. The source specified by MEASUREMENT:MEAS<x>:SOURCE1 must be selected for the measurement to be displayed. The source is selected using the SELECT:CH<x> command.

Group

Oscilloscope Measurement

Syntax

```
MEASUREMENT:MEAS<x>:STATE { OFF | ON | <NR1> }
```

```
MEASUREMENT:MEAS<x>:STATE?
```

Arguments

OFF or <NR1> = 0 turns measurements off. You can also turn the state off by deselecting the source.

ON or <NR1> \neq 0 turns measurements on.

Examples

```
MEASUREMENT:MEAS1:STATE ON  
turns measurement defined as MEAS1 on.
```

```
MEASUREMENT:MEAS4:STATE?  
returns either 0 or 1, indicating the state of MEAS4.
```

Related Commands

None

MEASUREMENT:MEAS<x>:TYPE

Sets or queries the oscilloscope measurement type for the measurement specified by <x>. This is equivalent to selecting the measurement in the Select Measurement pop-up menu.

Group

Oscilloscope Measurement

Syntax

MEASUREMENT:MEAS<x>:TYPE { AMPLitude | BURst | CMEan |
CRMs | FALL | FREQuency | HIGH | LOW | MAXimum | MEAN |
MINimum | NDUTy | NOVershoot | NWidth | PDUTy | PERIod |
PK2pk | POVershoot | PWidth | RISE | RMS }

MEASUREMENT:MEAS<x>:TYPE?

Arguments

AMPLitude is the high value minus the low value.

BURst is the time from the first MidRef crossing to the last MidRef crossing.

CMEan is the arithmetic mean over one cycle.

CRMs is the true Root Mean Square value over one waveform cycle.

FALL is the time that it takes for the falling edge of a pulse to fall from a HighRef value to a LowRef value.

FREQuency is the reciprocal of the period measured in Hertz.

HIGH is the 100% reference level.

LOW is the 0% reference level.

MAXimum is the highest amplitude.

MEAN is the arithmetic mean over the entire waveform.

MINimum is the lowest amplitude.

NDUTy is the ratio of the negative pulse width to the signal period expressed as a percentage.

NOVershoot is the negative overshoot, expressed as:

$$NOVershoot = 100 \times \frac{(Low - Minimum)}{Amplitude}$$

NWidth is the distance (time) between MidRef (usually 50%) amplitude points of a negative pulse.

PDUTy is the ratio of the positive pulse width to the signal period expressed as a percentage.

PERIod is the time, in seconds, it takes for one complete signal cycle to happen.

PK2pk is the absolute difference between the maximum and minimum amplitude.

POVershoot is the positive overshoot, expressed as:

$$POVershoot = 100 \times \frac{(Maximum - High)}{Amplitude}$$

PWIdth is the distance (time) between MidRef (usually 50%) amplitude points of a positive pulse.

RISe is the time that it takes for the leading edge of a pulse to rise from a low reference value to a high reference value.

RMS is the true Root Mean Square value.

Examples

MEASUREMENT:MEAS3:TYPE RMS

specifies MEAS3 to calculate the Root Mean Square value.

Related Commands

DMM:MEASUREMENT:MIN?, DMM:MEASUREMENT:MAX?,
DMM:MEASUREMENT:MAXMIN?, DMM:MEASUREMENT:AVG?,
DMM:MEASUREMENT:DELTA?

MEASUREMENT:MEAS<x>:UNITS? (Query Only)

Returns the units for the oscilloscope measurement specified by MEASUREMENT:MEAS<x>:TYPE.

Group

Oscilloscope Measurement

Syntax

MEASUREMENT:MEAS<x>:UNITS?

Returns

<QString> returns "V" for volts, "A" for amperes, "VV" for volts squared, "AA" for amperes squared, "W" for watts, "s" for seconds, "Hz" for hertz, or "%" for percent.

Examples

MEASUREMENT:MEAS3:UNITS?

might return "%", indicating the units for Oscilloscope Measurement 3 are percent.

Related Commands

None

MEASUREMENT:MEAS<x>:VALUE? (Query Only)

Returns the value that has been calculated for the oscilloscope measurement specified by <x>.

***NOTE.** This value is a display value and will be updated every 1/3 second.*

Group

Oscilloscope Measurement

Syntax

MEASUREMENT:MEAS<x>:VALUE?

Returns

<NR3>

Examples

MEASUREMENT:MEAS3:VALUE?

might return 28.75E6 if measurement number three is frequency.

Related Commands

DMM:MEASUREMENT:VALUE?

MEASUrement:METHOD

Sets or queries the method used to calculate the 0% and 100% reference level for the oscilloscope measurement. This is equivalent to setting the High-Low Setup in the Measure menu.

Group

Oscilloscope Measurement

Syntax

MEASUrement:METHOD { HISTogram | MINMax }

MEASUrement:METHOD?

Arguments

HISTogram sets the high and low waveform levels statistically using a histogram algorithm.

MINMax sets the high and low waveform levels to MAX and MIN respectively.

Examples

MEASUREMENT:METHOD HISTOGRAM

specifies that the high and low reference levels are set statistically.

MEASUREMENT:METHOD?

returns MINMAX when the reference levels are set to MAX and MIN.

Related Commands

None

MEASUrement:REFLevel? (Query Only)

Returns the oscilloscope measurement reference levels in percent and absolute terms.

Group

Oscilloscope Measurement

Syntax

MEASUrement:REFLevel?

Returns

None

Examples

MEASUrement:REFLevel?

might return the string :MEASUREMENT:REFLEVEL:METHOD

PERCENT;ABSOLUTE:HIGH 0.0E0;LOW 0.0E0;MID

0.0E0;;:MEASUREMENT:REFLEVEL:PERCENT:HIGH 9.0E1;LOW

1.0E1;MID 5.0E1

Related Commands

None

MEASUrement:REFLevel:ABSolute:HIGH

Sets or queries the high reference level for the oscilloscope measurement, and is the 100% reference level when MEASUrement:REFLevel:METHOD is set to ABSolute.

Group

Oscilloscope Measurement

Syntax

MEASUrement:REFLevel:ABSolute:HIGH <NR3>

MEASUrement:REFLevel:ABSolute:HIGH?

Arguments

<NR3> is the high reference level in vertical units. The default is 0.0.

Examples

MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH 1.71

sets the high reference level to 1.71 V.

Related Commands

None

MEASUREMENT:REFLevel:ABSolute:LOW

Sets or queries the low reference level for the oscilloscope measurement, and is the 0% reference level when MEASUREMENT:REFLevel:METHOD is set to ABSolute.

Group

Oscilloscope Measurement

Syntax

```
MEASUREMENT:REFLevel:ABSolute:LOW <NR3>
```

```
MEASUREMENT:REFLevel:ABSolute:LOW?
```

Arguments

<NR3> is the low reference level in vertical units. The default is 0.0.

Examples

```
MEASUREMENT:REFLEVEL:ABSOLUTE:LOW?  
might return 0.0E+0 as the low reference level.
```

Related Commands

None

MEASUREMENT:REFLevel:ABSolute:MID

Sets or queries the mid reference level for the oscilloscope measurement, and is the 50% reference level when MEASUREMENT:REFLevel:METHOD is set to ABSolute.

Group

Oscilloscope Measurement

Syntax

```
MEASUREMENT:REFLevel:ABSolute:MID <NR3>
```

```
MEASUREMENT:REFLevel:ABSolute:MID?
```

Arguments

<NR3> is the mid reference level in vertical units. The default is 0.0.

Examples

```
MEASUREMENT:REFLEVEL:ABSOLUTE:MID 0.71
```

sets the mid reference level to 0.71 volts.

Related Commands

None

MEASUREMENT:REFLEVEL:METHOd

Specifies which reference levels are used for oscilloscope measurement calculations.

Group

Oscilloscope 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. This method is not available using the front-panel buttons.

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?
```

returns either **ABSolute** or **PERCENT**, indicating the reference levels used.

Related Commands

None

MEASUREMENT:REFLevel:PERCent:HIGH

Sets or queries the percent, relative to HIGH, that is used to calculate the high reference level for oscilloscope measurements when MEASUREMENT:REFLevel:METHOD is set to PERCENT.

Group

Oscilloscope Measurement

Syntax

```
MEASUREMENT:REFLevel:PERCENT:HIGH <NR3>
```

```
MEASUREMENT:REFLevel:PERCENT:HIGH?
```

Arguments

<NR3> ranges from 0 to 100% and is the high reference level. The default is 90%. This control is not available using the front-panel buttons.

Examples

```
MEASUREMENT:REFLEVEL:PERCENT:HIGH 95
```

specifies that the high reference level is set to 95% of HIGH.

Related Commands

None

MEASUREMENT:REFLevel:PERCent:LOW

Sets or queries the percent, relative to LOW, that is used to calculate the low reference level for oscilloscope measurements when MEASUREMENT:REFLevel:METHOD is set to PERCENT.

Group

Oscilloscope Measurement

Syntax

MEASUREMENT:REFLevel:PERCent:LOW <NR3>

MEASUREMENT:REFLevel:PERCent:LOW?

Arguments

<NR3> ranges from 0 to 100% and is the low reference level. The default is 10%. This control is not available using the front-panel buttons.

Examples

MEASUREMENT:REFLEVEL:PERCENT:LOW?

might return 15, meaning that the low reference level is 15% of LOW.

Related Commands

None

MEASUREMENT:REFLevel:PERCent:MID

Sets or queries the percent, relative to MID, that is used to calculate the mid reference level for oscilloscope measurements when MEASUREMENT:REFLevel:METHod is set to PERCent.

Group

Oscilloscope Measurement

Syntax

MEASUREMENT:REFLevel:PERCent:MID <NR3>

MEASUREMENT:REFLevel:PERCent:MID?

Arguments

<NR3> ranges from 0 to 100% and is the mid reference level. The default is 50%. This control is not available using the front-panel buttons.

Examples

MEASUREMENT:REFLEVEL:PERCENT:MID 60

specifies that the mid reference level is set to 60% of MID.

Related Commands

None

NEWpass (No Query Form)

Changes the password that enables access to password protected data. The PASSWord command must be successfully executed before using this command or an execution error will be generated.

Group

Miscellaneous Oscilloscope

Syntax

NEWpass <QString>

Arguments

<QString> is the new password. The password can include up to 10 characters.

Examples

NEWPASS "mypassword"

creates a new password for accessing the user protected data.

Related Commands

PASSWord, *PUD

***OPC**

Generates the operation complete message in the Standard Event Status Register (SESR) when all pending operations finish. The *OPC? query places the ASCII character "1" into the output queue when all pending operations are finished. The *OPC? response is not available to read until all pending operations finish. For a complete discussion of the use of these registers and the output queue, see page 3-1.

Table 2–30 lists commands that generate an operation complete message.

Group

Status and Error

Syntax

*OPC

*OPC?

The *OPC command allows you to synchronize the operation of the instrument with your application program. Synchronization methods are described starting on page 3–9.

Table 2–30: Commands that Generate an Operation Complete Message

Operation	Command
Automatic scope adjustment	AUTOSet EXECute
Internal self-calibration	*CAL
Single sequence acquisition	ACQuire:STATE ON or ACQuire:STATE RUN (when ACQuire:STOPAfter is set to SE- Quence)
Hardcopy output	HARDCopy START

Arguments

None

Examples

*OPC?

might return a 1, which indicates that the operation is complete.

Related Commands

BUSY?, *WAI

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 instrument is powered off, or until the FACtory command, the PASSWord command with no arguments, or the *RST command is issued.

To change the password, you must first enter the valid password with the PASSWord command and then change to your new password with the NEWpass command. Remember that the password is case sensitive.

Group

Miscellaneous Oscilloscope

Syntax

PASSWord[<QString>]

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 *PUD and NEWpass set commands. You can still use the query version of *PUD.

Related Commands

NEWpass, *PUD

POWER:BACKlight

Sets and queries the setting of the backlight timer. This timer, which automatically turns off the backlight after a period of user inactivity, is disabled when the instrument is operating with the AC adapter.

Group

Miscellaneous Oscilloscope

Syntax

```
POWER:BACKlight { <NR1> | INFinite }
```

```
POWER:BACKlight?
```

Arguments

<NR1> is the timer setting in minutes in the range from 1 through 15.

INFinite disables the timer.

Examples

```
POWER:BACKlight 4
```

sets the timer to turn off the backlight after four minutes of user inactivity.

Related Commands

None

POWER:STANDbY

Sets and queries the setting of the automatic power off timer. This timer, which automatically turns off the instrument after a period of user inactivity, is disabled when the instrument is operating with the AC adapter.

Group

Miscellaneous Oscilloscope

Syntax

```
POWER:STANDbY { <NR1> | INFinite }
```

POWER:STANdby?

Arguments

<NR1> is the timer setting in minutes in the range from 1 through 15.

INFinite disables the timer.

Examples

POWER:STANdby 6

sets the timer to turn off the instrument after six minutes of user inactivity.

Related Commands

None

*PSC

Sets and queries the power-on status flag that controls the automatic power-on handling of the DESER, SRER, and ESER registers. When *PSC is true, the DESER register is set to 255 and the SRER and ESER registers are set to 0 at power on. When *PSC is false, the current values in the DESER, SRER, and ESER registers are preserved in nonvolatile memory when power is shut off and are restored at power on. For a complete discussion of the use of these registers, see page 3–1.

Group

Status and Error

Syntax

*PSC <NR1>

*PSC?

Arguments

<NR1> = 0 sets the power-on status clear flag to false, disables the power on clear, and allows the instrument to possibly assert SRQ after power on.

<NR1> \neq 0 sets the power-on status clear flag true. Sending *PSC 1, therefore, enables the power-on status clear and prevents any SRQ assertion after power on. Using an out-of-range value causes an execution warning.

Examples

*PSC 0

sets the power-on status clear flag to false.

*PSC?

might return the value 1, showing that the power-on status clear flag is set to true.

Related Commands

DESE, *ESE, FACtory, *RST, *SRE

*PUD

Sets or queries a string of Protected User Data. This data is protected by the PASSWord command. You can modify it only by first entering the correct password. The password is not necessary to query the data.

Group

Miscellaneous Oscilloscope

Syntax

*PUD <Block>

*PUD?

Arguments

<Block> is a string containing up to 100 characters.

Examples

*PUD #229This instrument belongs to me
stores the string “This instrument belongs to me” in the user protected data area.

*PUD?
might return #221Property of Company X.

Related Commands

PASSWord

***RCL (No Query Form)**

Restores the state of the instrument from a copy of its settings stored in memory. (The settings are stored using the *SAV command.) This command is equivalent to RECALL:SETUp, and performs the same function as the Recall Saved Setup item in the front-panel Save/Recall Setup menu.

Group

Save and Recall

Syntax

*RCL <NR1>

Arguments

<NR1> is a value in the range from 1 to 10, and specifies a setup storage location. Using an out-of-range value causes an execution error (222, “Data out of range”).

Examples

*RCL 3
restores the instrument from a copy of the settings stored in memory location 3.

Related Commands

FACTory, *LRN?, RECALL:SETUp, *RST, *SAV, SAVe:SETUp

RECALL:SETUp (No Query Form)

Restores a stored or factory front-panel setup of the instrument. This command is equivalent to selecting Recall Saved Setup or Recall Factory Setup in the Save/Recall Setup menu.

Group

Save and Recall

Syntax

```
RECALL:SETUp { FACTory | <NR1> }
```

Arguments

FACTory selects the factory setup.

<NR1> is a value in the range from 1 to 10 and specifies a setup storage location. Using an out-of-range value causes an execution error (222, “Data out of range”).

Examples

```
RECALL:SETUP FACTORY
```

recalls the front-panel setup to its factory defaults.

Related Commands

FACTory, *RCL, *RST, *SAV, SAVe:SETUp

REFA:WAVEform

This command maps a stored waveform to the Ref A waveform.

Group

Save and Recall

Syntax

```
REFA:WAVEform STO[1-10]
```

```
REFA:WAVEform?
```

Arguments

STO[1-10] selects a stored waveform location in the range from 1 through 10.

Examples

REFA:WAVEform ST06
maps stored waveform six to Ref A.

Related Commands

REFB:WAVEform

REFB:WAVEform

This command maps a stored waveform to the Ref B waveform.

Group

Save and Recall

Syntax

REFB:WAVEform STO[1-10]

REFB:WAVEform?

Arguments

STO[1-10] selects a stored waveform location in the range from 1 through 10.

Examples

REFA:WAVEform ST03
maps stored waveform three to Ref B.

Related Commands

REFA:WAVEform

REM (No Query Form)

Specifies a comment. This line is ignored by the instrument.

Group

Miscellaneous Oscilloscope

Syntax

REM <QString>

Arguments

<QString> is a string that can have a maximum of 80 characters.

Examples

REM "This is a comment"
is ignored by the instrument.

Related Commands

None

RS232? (Query Only)

Queries the RS232 settings.

Group

Miscellaneous Oscilloscope

Syntax

RS232?

Returns

None

Examples

RS232?
might return: RS232:BAUD 9600;MODE RAW;PACE XON;PARITY
NONE;SBITS 2;CONTROL:RTS RFR;RS232:TRANSMIT;TERMINATOR
LF;DELAY 0.0

Related Commands

RS232: BAUD, RS232: HARDFLAGGING, RS232: PARITY,
RS232:SOFTFLAGGING, RS232: STOPBITS

RS232:BAUd

Sets or queries the RS-232 interface transmission speed. If no flow control (flagging) is used, commands may be received faster than the oscilloscope can process them. Also, if another command is sent immediately after this command, without waiting for the baud rate to be programmed, the first couple of characters may be lost.

Group

Miscellaneous Oscilloscope

Syntax

RS232:BAUd <NR1>

RS232:BAUd?

Arguments

<NR1> where <NR1> can be 300, 600, 1200, 2400, 4800, 9600, 19200, or 38400.

Examples

RS232:BAUD 9600

sets the transmission rate to 9600 baud.

Related Commands

RS232: HARDFLAGGING, RS232: PARITY, RS232:SOFTFLAGGING,
RS232: STOPBITS, RS232?

RS232:CONTROL:DCD

The instrument accepts but ignores this command.

Group

Miscellaneous Oscilloscope

Syntax

```
RS232:CONTROL:DCD { OFF | 0 }
```

```
RS232:CONTROL:DCD?
```

Arguments

OFF or 0 turns off DCD monitoring.

Examples

```
RS232:CONTROL:DCD OFF
```

turns off DCD monitoring. This is essentially a no-op since DCD monitoring is always off.

Related Commands

RS232: HARDFLAGGING, RS232: PARITY, RS232:SOFTFLAGGING, RS232: STOPBITS, RS232?

RS232:CONTROL:RTS

Sets or queries the state of RS232 hard flagging. This command performs the same function as RS232:HARDFlagging.

Group

Miscellaneous Oscilloscope

Syntax

```
RS232:CONTROL:RTS { ON | RFR | IBFull }
```

```
RS232:CONTROL:RTS?
```

Arguments

<ON> asserts RTS (Request to Send).

<RFR> enables hard flagging.

<IBFull> enables hard flagging.

Examples

```
RS232:CONTROL:RTS RFR
```

enables hard flagging.

Related Commands

RS232: HARDFLAGGING, RS232:SOFTFLAGGING, RS232:STOPBITS, RS232?

RS232:HARDFlagging

Sets or queries the state of RS232 hard flagging. When hard flagging is enabled, the instrument sends data as long as CTS (Clear To Send) is asserted. When receiving data, the instrument asserts RTS (Request To Send) until the input buffer is almost full. When the instrument no longer asserts RTS, it continues to read incoming data until the input buffer is full and then reports an input overrun error. The instrument asserts DTR (Data Terminal Ready) at all times when the instrument power is on.

Group

Miscellaneous Oscilloscope

Syntax

RS232:HARDFlagging { ON | OFF | <NR1> }

RS232:HARDFlagging?

Arguments

<ON> or <NR1> \neq 0 turn on hard flagging.

<OFF> or <NR1> = 0 turn off hard flagging (RTS always asserted).

Examples

RS232:HARDFLAGGING ON
turns on hard flagging.

Related Commands

RS232: BAUD, RS232: PARITY, RS232:SOFTFLAGGING, RS232:STOPBITS, RS232?

RS232:MODE

The instrument accepts but ignores this command. The query always returns RAW.

Group

Miscellaneous Oscilloscope

Syntax

RS232:MODE RAW

RS232:MODE?

Arguments

RAW (GPIB emulation) mode.

Examples

RS232:MODE?

always returns RAW.

Related Commands

RS232: BAUD, RS232: HARDFLAGGING, RS232: PARITY, RS232:SOFTFLAGGING, RS232?

RS232:PACE

Sets or queries the input and output soft flagging over the RS-232 port. This command performs the same function as RS232:SOFT-Flagging.

Group

Miscellaneous Oscilloscope

Syntax

RS232:PACE { XON | NONE }

RS232:PACE?

Arguments

<XON> turn on soft flagging.

<NONE> turn off soft flagging.

Examples

RS232:PACE XON
turns on soft flagging.

Related Commands

RS232: BAUD, RS232: HARDFLAGGING, RS232: PARITY,
RS232: STOPBITS, RS232?

RS232:PARity

Sets or queries the parity used for all RS-232 data transfers. When parity is odd or even, the instrument generates the selected parity on output and checks all input against the selected parity. When parity is none, the instrument performs no input parity error checks and generates no output parity. When the parity (9th) bit does not match the parity type, the instrument reports a parity error. If another command is sent immediately after this command, without waiting for the parity to be programmed, the first few characters may be lost.

Group

Miscellaneous Oscilloscope

Syntax

RS232:PARity { EVEN | ODD | NONE }

RS232:PARity?

Arguments

<EVEN> sets even parity.

<ODD> sets odd parity.

<NONE> sets no parity (no ninth bit transmitted).

Examples

RS232:PARITY EVEN
sets even parity.

Related Commands

RS232: BAUD, RS232: HARDFLAGGING, RS232:SOFTFLAGGING, RS232: STOPBITS, RS232?

RS232:PRESEt (No Query Form)

Sets RS-232 parameters to default values. The RS232? query will show the new settings.

```
RS232:MODE RAW
RS232:CONTRol:RTS RFR
RS232:CONTRol:DCD OFF
RS232:PACe NONE
RS232:BAUD 9600
RS232:PARity NONE
RS232:SBITs 1
RS232:TRANsmit:DELay 0
```

Group

Miscellaneous Oscilloscope

Syntax

RS232:PRESEt

Arguments

None

Examples

RS232:PRESEt
sets RS232 parameters to the default values.

Related Commands

RS232: BAUD, RS232: HARDFLAGGING, RS232:SOFTFLAGGING, RS232: STOPBITS, RS232?

RS232:SBITS

Sets or queries the number of transmission stop bits sent with each character to identify the end of data for that character. This command performs the same function as RS232:STOPBits.

Group

Miscellaneous Oscilloscope

Syntax

RS232:SBITS <NR1>

RS232:SBITS?

Arguments

<NR1> where <NR1> can either be 1 or 2.

Examples

RS232:SBITS 1
sets the number of stop bits to 1.

Related Commands

RS232:BAUD, RS232:HARDFLAGGING, RS232:PARITY,
RS232:SOFTFLAGGING, RS232?

RS232:SOFTFlagging

Sets or queries the input and output soft flagging over the RS-232 port. After receiving an XOFF (DC3), the instrument sends two or less characters. The instrument sends an XOFF character when its input buffer is running out of space. After sending an XOFF character it can receive at least 20 more bytes. The instrument begins transmitting data again when it receives an XON (DC1) character. It sends XON when its input buffer has an acceptable number of free bytes.

When soft flagging is enabled and binary data is transferred, data transmission will lock up if the data contains XOFF or XON characters.

Group

Miscellaneous Oscilloscope

Syntax

```
RS232:SOFTFlagging { ON | OFF | <NR1> }
```

```
RS232:SOFTFlagging?
```

Arguments

<ON> or <NR1> \neq 0 turns on soft flagging.

<OFF> or <NR1> = 0 turns off soft flagging.

Examples

```
RS232:SOFTFLAGGING ON  
turns on soft flagging.
```

Related Commands

```
RS232: BAUD, RS232: HARDFLAGGING, RS232: PARITY,  
RS232: STOPBITS, RS232?
```

RS232:STOPBits

Sets or queries the number of transmission stop bits sent with each character to identify the end of data for that character. The standard setting for most computer equipment is 1 stop bit. If another command is sent immediately after this command without waiting for it to complete, the first couple of characters may be lost.

Group

Miscellaneous Oscilloscope

Syntax

```
RS232:STOPBits <NR1>
```

RS232:STOPBits?

Arguments

<NR1> where <NR1> can either be 1 or 2.

Examples

RS232:STOPBITS 1
sets the number of stop bits to 1.

Related Commands

RS232:BAUD, RS232:HARDFLAGGING, RS232:PARITY,
RS232:SOFTFLAGGING, RS232?

RS232:TRANsmit:DELay

Sets or queries the minimum amount of time to wait after receiving a query command before sending the response. This is provided for old terminals and computers that cannot accept data immediately after sending data.

Group

Miscellaneous Oscilloscope

Syntax

RS232:TRANsmit:DELay { <NR1> }

RS232:TRANsmit:DELay?

Arguments

<NR1> the delay value from 0 s to 60 s.

Examples

RS232:TRANsmit:DELay 0
sets the transmit delay to 0 s.

Related Commands

RS232:HARDFLAGGING, RS232:SOFTFLAGGING, RS232:
STOPBITS, RS232?

RS232:TRANsmit:TERMinator

Sets or queries the end-of-line (EOL) terminator. When transmitting, the instrument appends the terminator to the end of each message. When receiving, the instrument accepts all four terminators, regardless of the currently selected terminator. When a combination of multiple characters is selected (CRLF or LFCR), the instrument interprets the first character as the terminator; it treats the second character as a null command.

CR represents an ASCII carriage return character (0x0D) and LF represents an ASCII linefeed character (0x0A).

Group

Miscellaneous Oscilloscope

Syntax

RS232:TRANsmit:TERMinator { CR | LF | CRLf | LFCr }

RS232:TRANsmit:TERMinator?

Arguments

<CR> selects the carriage return character as the EOL terminator.

<LF> selects the line feed character as the EOL terminator.

<CRLf> selects the carriage return and line feed characters as the EOL terminator.

<LFCr> selects the line feed and carriage return characters as the EOL terminator.

Examples

RS232:TRANsmit:TERMinator CR
sets the carriage return as the EOL terminator.

Related Commands

RS232: HARDFLAGGING, RS232:SOFTFLAGGING, RS232:STOPBITS, RS232?

***RST (No Query Form)**

(Reset) Returns the instrument to a known set of instrument settings, but does not purge any aliases or stored settings. This command is identical to the FACtory command.

Group

Status and Error

Syntax

*RST

Setting the instrument to factory default has the following impact on the programming interface:

- Clears the Event Status Enable Register
- Clears the Service Request Enable Register
- Sets the Device Event Status Enable Register to 255
- Sets the Power On Status Clear Flag to TRUE
- Purges all defined aliases
- Enables all Command Headers (HEADer ON)
- Sets the macro defined by *DDT to a “zero-length field”
- Clears the pending operation flag and associated operations

The *RST command does not alter the following items:

- The state of the RS-232 interface
- Calibration data that affects device specifications
- Protected user data
- Stored settings
- Stored waveforms or data
- The current password (if implemented)
- Hard copy parameters

Arguments

None

Examples

None

Related Commands

FACtory, *PSC, *RCL, RECALL:SETUp, *SAV, SAVE:SETUp

*SAV (No Query Form)

(Save) Stores the state of the instrument into a specified memory location. You can later use the *RCL command to restore the instrument to this saved state. This is equivalent to selecting the Save Current Setup in the Save/Recall Setup menu.

Group

Save and Recall

Syntax

*SAV <NR1>

Arguments

<NR1> is a value in the range from 1 to 10 and specifies a memory location. Using an out-of-range value causes an execution error. Any settings that have been stored previously at this location will be overwritten.

Examples

*SAV 2

saves the current settings in memory location 2.

Related Commands

FACtory, *RCL, RECALL:SETUp, SAVE:SETUp

SAVe:SETUp (No Query Form)

Saves the current front-panel setup into the specified memory location. This is equivalent to selecting the Save Current Setup in the Save/Recall Setup menu.

Group

Save and Recall

Syntax

SAVe:SETUp <NR1>

Arguments

<NR1> is a value in the range from 1 to 10 and specifies a memory location. 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 front-panel setup in memory location 5.

Related Commands

RECALL:SETUp, *RCL, *SAV

SAVe:WAVEform (No Query Form)

Stores an oscilloscope waveform in one of ten stored waveform memory locations.

Group

Save and Recall

Syntax

SAVe:WAVEform <wfm><Comma>STO[1-10]

Arguments

<wfm> is CH1, CH2, MATH, MATH1, REFA, or REFB. This is the waveform that will be saved.

STO[1-10] is the location, in the range from 1 through 10, where the waveform will be stored.

Examples

```
SAVE:WAVEFORM MATH,STO4
```

saves the math waveform in stored waveform memory location four.

Related Commands

```
DMM:SAVe:DATA
```

SElect? (Query Only)

Returns the selected oscilloscope waveform and the display status of all waveforms.

Group

Vertical

Syntax

SElect?

Returns

None

Examples

SELECT?

might return :SELECT:CH1 1;CH2 0;MATH 0;REFA 0;
REFB 0;CONTROL CH1

Related Commands

None

SElect:<wfm>

Controls the display and selection of oscilloscope waveforms. There can be up to five waveforms displayed at one time, but only one waveform can be selected at a time. The selected waveform is the waveform that was most recently turned on. This command is equivalent to pressing a front-panel channel button (CH 1, CH 2, MATH, REF A, or REF B). In this command, <wfm> can be CH1, CH2, MATH, REFA, or REFB.

Group

Vertical

Syntax

SElect:<wfm> { OFF | ON | <NR1> }

SElect:<wfm>?

Arguments

OFF or <NR1> = 0 turns off the display of the specified waveform.

ON or <NR1> \neq 0 turns on the display of the specified waveform. The waveform also becomes the selected waveform.

Examples

```
SELECT:CH2 ON
```

turns the channel 2 display on and selects channel 2.

```
SELECT:REFA?
```

returns either 0 or 1, indicating whether the REFA waveform is displayed.

Related Commands

None

SElect:CONTROL

Sets or queries the oscilloscope waveform that is currently measured and adjusted by the cursor and vertical commands.

Group

Vertical

Syntax

```
SElect:CONTROL <wfm>
```

```
SElect:CONTROL?
```

Arguments

<wfm> is CH<x>, MATH, or REF<x>, and is the selected waveform.

Examples

```
SELECT:CONTROL?
```

might return CH1 as the selected waveform.

Related Commands

None

SET? (Query Only)

Returns a string listing the instrument settings except for the calibration values. You can use this string to return the instrument to the state it was in when you sent SET?. This command is identical to the *LRN? command.

Group

Miscellaneous Oscilloscope

Syntax

SET?

***NOTE.** The SET? query always returns a string with command headers, regardless of the setting of the HEADer command. This is because the returned string is intended to be able to be sent back to the instrument as a command string. The VERBose command can still be used to specify whether the returned headers should be abbreviated or full length.*

Returns

None

Examples

SET?

a partial return string may look like this:

```
:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV
8;NUMAVG 16;:HEADER 1;:VERBOSE 1; :ALIAS:STATE 0;
:DISPLAY:FORMAT YT;STYLE VECTORS;PERSISTENCE
500.0E-3;GRATICULE FULL;TRIGT 1;CONTRAST 50;:LOCK NONE;
:HARDCOPY:FORMAT EPSIMAGE;PORT RS232;LAYOUT PORTRAIT;
```

Related Commands

HEADer, *LRN?, VERBose

***SRE**

(Service Request Enable) sets and queries the bits in the Service Request Enable Register (SRER). For a complete discussion of the use of these registers, see page 3–1.

Group

Status and Error

Syntax

*SRE <NR1>

*SRE?

Arguments

<NR1> is a value in the range from 0 to 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 its value through a power cycle.

Examples

*SRE 48

sets the bits in the SRER to 00110000 binary.

*SRE?

might return a value of 32, showing that the bits in the SRER have the binary value 00100000.

Related Commands

*CLS, DESE, *ESE, *ESR?, EVENT?, EVMSg?, FACtory, *PSC, *STB?

***STB? (Query Only)**

(Read Status Byte) query returns the contents of the Status Byte Register (SBR) using the Master Summary Status (MSS) bit. For a complete discussion of the use of these registers, see page 3–1.

Group

Status and Error

Syntax

*STB?

Returns

<NR1>

Examples

*STB?

might return the value 96, showing that the SBR contains the binary value 01100000.

Related Commands

*CLS, DESE, *ESE, *ESR?, EVENT?, EVMSg?, FACtory, *SRE

TEKSecure

Initializes waveform, data, and setup memories. This command overwrites any previously stored data, nulls all waveform reference memory, and puts all setups in the factory init state. It then verifies that the waveform, data, and setup memories are initialized and displays an on-screen pass or a fail notifier on completion.

Group

Miscellaneous Oscilloscope

Syntax

TEKSecure

Arguments

None

Examples

None

Related Commands

None

***TRG (No Query Form)**

(Trigger) Executes commands that are defined by *DDT.

Group

Miscellaneous Oscilloscope

Syntax

*TRG

Arguments

None

Examples

*TRG

immediately executes all commands that have been defined by *DDT.

Related Commands

Alias commands, *DDT

TRIGger

Forces an oscilloscope trigger event to occur. When used as a query, TRIGger? returns the current trigger parameters.

Group

Trigger

Syntax

TRIGger FORCe

TRIGger?

Arguments

FORCe creates a trigger event. If TRIGger:STATE is REAdy, the acquisition will complete; otherwise this command will be ignored. This is equivalent to selecting FORCE TRIGGER in the acquire menu.

Examples

TRIGGER FORCe

forces a trigger event to occur.

TRIGGER?

might return :TRIGGER:MAIN:MODE AUTO;TYPE EDGE;LEVEL
-4.0E0;HOLDOFF:VALUE 4.95E-7;:TRIGGER:MAIN:EDGE:SOURCE
CH1;COUPLING DC;SLOPE RISE;:TRIGGER:MAIN:PULSE:WHEN
LESS;EQUAL 1.0E1;NOTEQUAL 1.0E1;HOLDOFF:VALUE
4.95E-7;:TRIGGER:MAIN:PULSE:SOURCE CH1;POLARITY
NEGATIVE;WIDTH 9.9E-8;:TRIGGER:MAIN:VIDEO:SOURCE
CH1;HOLDOFF:VALUE 4.95E-7;:TRIGGER:MAIN:VIDEO:FIELD
FIELD1;SCAN RATE1

Related Commands

None

TRIGger:MAIn

Sets the oscilloscope main trigger level and returns the current main trigger parameters.

Group

Trigger

Syntax

TRIGger:MAIn SETLeve1

TRIGger:MAIn?

Arguments

SETLevel sets the main trigger level to half way between the MIN and MAX amplitudes of the trigger source input. This is equivalent to pressing the front-panel SET LEVEL TO 50% button.

Examples

```
TRIGGER:MAIN SETLEVEL
```

sets the main trigger level mid way between MAX and MIN.

Related Commands

None

TRIGger:MAIn:EDGE? (Query Only)

Returns the oscilloscope trigger coupling, source, and slope settings for the main edge trigger.

Group

Trigger

Syntax

```
TRIGger:MAIn:EDGE?
```

Returns

None

Examples

```
TRIGGER:MAIN:EDGE?
```

might return SOURCE CH1;COUPLING DC;SLOPE RISE

Related Commands

None

TRIGger:MAIn:EDGE:COUPling

Sets or queries the type of coupling for the oscilloscope main edge trigger. This is equivalent to setting Coupling in the Edge trigger menu.

Group

Trigger

Syntax

```
TRIGger:MAIn:EDGE:COUPling { DC | HFRej | LFRej |  
NOISErej }
```

```
TRIGger:MAIn:EDGE:COUPling?
```

Arguments

DC selects DC trigger coupling.

HFRej coupling removes the high-frequency components of the DC signal.

LFRej coupling removes the low-frequency components of the AC signal.

NOISErej selects DC low sensitivity. It requires added signal amplitude for more stable, less false triggering.

Examples

```
TRIGGER:MAIN:EDGE:COUPLING DC  
sets the main edge trigger coupling to DC.
```

Related Commands

None

TRIGger:MAIn:EDGE:SLOpe

Selects a rising or falling slope for the oscilloscope main edge trigger. This is equivalent to setting Slope in the Edge trigger menu.

Group

Trigger

Syntax

```
TRIGger:MAIn:EDGE:SLOpe { FALL | RISE }
```

```
TRIGger:MAIn:EDGE:SLOpe?
```

Arguments

FALL specifies to trigger on the falling or negative edge of a signal.

RISe specifies to trigger on the rising or positive edge of a signal.

Examples

```
TRIGGER:MAIN:EDGE:SLOPE RISE
```

sets the main edge trigger to occur on the rising slope.

Related Commands

None

TRIGger:MAIn:EDGE:SOUrce

Sets or queries the source for the oscilloscope main edge trigger. This is equivalent to setting Source in the Edge trigger menu.

Group

Trigger

Syntax

```
TRIGger:MAIn:EDGE:SOUrce { CH<x> }
```

```
TRIGger:MAIn:EDGE:SOUrce?
```

Arguments

CH<x> specifies one of the input channels.

Examples

TRIGGER:MAIN:EDGE:SOURCE CH1
specifies the channel 1 as the edge trigger source.

TRIGGER:MAIN:EDGE:SOURCE?
might return CH2 for the edge trigger source.

Related Commands

None

TRIGger:MAIn:HOLdoff? (Query Only)

Returns the oscilloscope main trigger holdoff value.

Group

Trigger

Syntax

TRIGger:MAIn:HOLdoff?

Returns

None

Examples

TRIGGER:MAIN:HOLDOFF?
might return :TRIGGER:MAIN:HOLDOFF:VALUE 5.0E-7.

Related Commands

None

TRIGger:MAIn:HOLdoff:VALue

Sets or queries the oscilloscope main trigger holdoff value.

Group

Trigger

Syntax

TRIGger:MAIn:HOLdoff:VALue <NR3>

TRIGger:MAIn:HOLdoff:VALue?

Arguments

<NR3> is the holdoff, from 495 ns to 10 s.

Examples

TRIGGER:MAIN:HOLDOFF:VALUE 10
sets the holdoff value to 10 s.

Related Commands

None

TRIGger:MAIn:LEVel

Sets the oscilloscope main trigger level. This command is equivalent to adjusting the front-panel TRIGGER LEVEL rocker.

Group

Trigger

Syntax

TRIGger:MAIn:LEVel { ECL | TTL | <NR3> }

TRIGger:MAIn:LEVel?

Arguments

ECL specifies a preset ECL level of -1.3 V.

TTL specifies a preset TTL level of 1.4 V.

<NR3> specifies the main trigger level, in vertical units.

Examples

TRIGGER:MAIN:LEVEL?
might return 1.4, indicating that the main edge trigger is set to 1.4 V.

Related Commands

None

TRIGger:MAIn:MODE

Sets or queries the oscilloscope trigger mode.

Group

Trigger

Syntax

```
TRIGger:MAIn:MODE { AUTO | NORMa1 }
```

```
TRIGger:MAIn:MODE?
```

Arguments

AUTO generates a trigger if a trigger is not detected within a specific time period. AUTO also enables roll mode 500 ms/div and slower sweep speeds.

NORMa1 waits for a valid trigger event.

Examples

```
TRIGGER:MAIN:MODE AUTO
```

specifies that a trigger event is automatically generated.

Related Commands

None

TRIGger:MAIn:PULse:EQUAL

Sets and queries the oscilloscope pulse-width trigger equality tolerance. The oscilloscope triggers on pulses equal to the width setting plus or minus this tolerance. This command is equivalent to setting the Equal To tolerance in the Pulse trigger menu.

Group

Trigger

Syntax

TRIGger:MAIn:PULse:EQUAL { 5 | 10 | 15 | 20 }
TRIGger:MAIn:PULse:EQUAL?

Arguments

5 specifies $\pm 5\%$ tolerance
10 specifies $\pm 10\%$ tolerance
15 specifies $\pm 15\%$ tolerance
20 specifies $\pm 20\%$ tolerance

Examples

TRIGger:MAIn:PULse:EQUAL 15
sets the tolerance to $\pm 15\%$

Related Commands

TRIGger:MAIn:PULse:WIDth, TRIGger:MAIn:PULse:POLarity,
TRIGger:MAIn:PULse:WHEn

TRIGger:MAIn:PULse:HOLDOff

Group

Trigger

Syntax

TRIGger:MAIn:PULse:HOLdoff?

Returns

None

Examples

TRIGGER:MAIN:PULse:HOLDOFF?
might return :TRIGGER:MAIN:PULse:HOLDOFF:VALUE 5.0E-7.

Related Commands

None

TRIGger:MAIn:PULse:HOLDOff:VALue

Sets or queries the oscilloscope main trigger holdoff value.

Group

Trigger

Syntax

```
TRIGger:MAIn:PULse:HOLdoff:VALue <NR3>
```

```
TRIGger:MAIn:PULse:HOLdoff:VALue?
```

Arguments

<NR3> is the holdoff, from 495 ns to 10 s.

Examples

```
TRIGGER:MAIN:PULSE:HOLDOFF:VALUE 10
```

sets the holdoff value to 10 s.

Related Commands

None

TRIGger:MAIn:PULse:NOTEqual

Sets and queries the oscilloscope pulse-width trigger inequality tolerance. The oscilloscope triggers on all pulses except those equal to the width setting plus or minus this tolerance. This command is equivalent to setting the Not Equal To tolerance in the Pulse trigger menu.

Group

Trigger

Syntax

```
TRIGger:MAIn:PULse:NOTEqual { 5 | 10 | 15 | 20 }
```

```
TRIGger:MAIn:PULse:NOTEqual?
```

Arguments

5 specifies $\pm 5\%$ tolerance

10 specifies $\pm 10\%$ tolerance

15 specifies $\pm 15\%$ tolerance

20 specifies $\pm 20\%$ tolerance

Examples

TRIGger:MAIn:PULse:NOTEqual 15
sets the tolerance to $\pm 15\%$

Related Commands

TRIGger:MAIn:PULse:WIDth, TRIGger:MAIn:PULse:POLarity,
TRIGger:MAIn:PULse:WHEn

TRIGger:MAIn:PULse:POLarity

Sets and queries the oscilloscope pulse-width trigger polarity. This command is equivalent to setting the Polarity in the Pulse trigger menu.

Group

Trigger

Syntax

TRIGger:MAIn:PULse:POLarity { POSitive | NEGative }
TRIGger:MAIn:PULse:POLarity?

Arguments

POSitive specifies positive polarity.

NEGative specifies negative polarity.

Examples

TRIGger:MAIn:PULse:POLarity NEGative
sets the polarity to negative.

Related Commands

TRIGger:MAIn:PULse:WIDth, TRIGger:MAIn:PULse:WHEn

TRIGger:MAIn:PULse:SOUrce

Sets or queries the source for the oscilloscope pulse trigger. This command is equivalent to setting the Source in the Pulse trigger menu.

Group

Trigger

Syntax

```
TRIGger:MAIn:PULse:SOUrce { CH<x> }
```

```
TRIGger:MAIn:PULse:SOUrce?
```

Arguments

CH<x> specifies one of the input channels.

Examples

```
TRIGGER:MAIN:PULSE:SOURCE CH1
```

specifies channel 1 as the pulse trigger source.

```
TRIGGER:MAIN:PULSE:SOURCE?
```

might return CH2 for the pulse trigger source.

Related Commands

None

TRIGger:MAIn:PULse:WHEn

Sets and queries the oscilloscope pulse-width trigger condition. This command is equivalent to setting the trigger condition in the Pulse trigger menu.

Group

Trigger

Syntax

```
TRIGger:MAIn:PULse:WHEn { LESS | GREATER | EQUAL |  
NOTEQUAL }  
TRIGger:MAIn:PULse:WHEn?
```

Arguments

LESS specifies triggering on a pulse width less than the width setting.

GREATER specifies triggering on a pulse width greater than the width setting.

EQUAL specifies triggering on a pulse width equal to the width setting plus or minus the tolerance setting.

NOTEQUAL specifies triggering on a pulse width not equal to the width setting plus or minus the tolerance setting.

Examples

```
TRIGger:MAIn:PULse:WHEn LESS  
specifies triggering on a pulse width less than the width setting.
```

Related Commands

TRIGger:MAIn:PULse:POLarity, TRIGger:MAIn:PULse:WHEn,
TRIGger:MAIn:PULse:EQUAL, TRIGger:MAIn:PULse:NOTEQUAL

TRIGger:MAIn:PULse:WIDth

Sets and queries the oscilloscope pulse width setting. This command is equivalent to setting the Width in the Pulse trigger menu.

Group

Trigger

Syntax

```
TRIGger:MAIn:PULse:WIDth <NR3>  
TRIGger:MAIn:PULse:WIDth?
```

Arguments

NR3 is the pulse width, from 99ns to 1 s. Resolution of the setting is 33 ns or approximately 1% of the setting (whichever is greater). Values not in the sequence are set to the nearest allowed value.

Examples

TRIGger:MAIn:PULse:WIDth 9.9E-7
sets the pulse width to 990 ns

Related Commands

TRIGger:MAIn:PULse:POLarity, TRIGger:MAIn:PULse:WHEn

TRIGger:MAIn:TYPe

Sets or queries the type of oscilloscope trigger. This is equivalent to setting Type in the Trigger menu.

Group

Trigger

Syntax

TRIGger:MAIn:TYPe { EDGE | PULse | VIDEo }

TRIGger:MAIn:TYPe?

Arguments

EDGE is a normal trigger. A trigger event occurs when a signal passes through a specified voltage level in the specified direction and is controlled by the TRIGger:MAIn:EDGE commands.

PULse specifies that a trigger occurs when the signal is less than, greater than, equal to, or not equal to a specific pulse width.

VIDEo specifies that a trigger occurs when a specified signal is found and is controlled by the TRIGger:MAIn:VIDEo commands.

Examples

TRIGGER:MAIN:TYPE?
might return VIDEO indicating that the main trigger type is a video trigger.

Related Commands

None

TRIGger:MAIn:VIDeo:FIELD

Sets or queries the field the oscilloscope video trigger detects.

Group

Trigger

Syntax

```
TRIGger:MAIn:VIDeo:FIELD { FIELD1 | FIELD2 | ANYField |  
ANY }
```

```
TRIGger:MAIn:VIDeo:FIELD?
```

Arguments

FIELD1 specifies interlaced video field 1.

FIELD2 specifies interlaced video field 2.

ANYField specifies any field.

ANY specifies any line.

Examples

```
TRIGGER:MAIN:VIDEO:FIELD1  
selects field 1.
```

Related Commands

None

TRIGger:MAIn:VIDeo:HOLdoff? (Query Only)

Returns the oscilloscope video trigger holdoff value.

Group

Trigger

Syntax

```
TRIGger:MAIn:VIDeo:HOLdoff?
```

Returns

None

Examples

TRIGGER:MAIN:VIDEO:HOLDOFF?

might return :TRIGGER:MAIN:VIDEO:HOLDOFF:VALUE 5.0E-7.

Related Commands

None

TRIGger:MAIn:VIDeo:HOLdoff:VALue

Sets or queries the oscilloscope video trigger holdoff value. This is equivalent to setting Holdoff in the Video trigger menu.

Group

Trigger

Syntax

TRIGger:MAIn:VIDeo:HOLdoff:VALue <NR3>

TRIGger:MAIn:VIDeo:HOLdoff:VALue?

Arguments

<NR3> is the holdoff, from 495 ns to 10 s.

Examples

TRIGger:MAIn:VIDeo:HOLdoff:VALue 3E-03

set the holdoff value to 3 ms.

Related Commands

None

TRIGger:MAIn:VIDeo:SCAN

Sets or queries the video trigger scan rate range. This is equivalent to setting the Scan Rate in the Video trigger menu.

Group

Trigger

Syntax

```
TRIGger:MAIn:VIDeo:SCAN { RATE1 | RATE2 | RATE3 | RATE4  
| RATE5 }
```

```
TRIGger:MAIn:VIDeo:SCAN?
```

Arguments

RATE1 specifies a 15 kHz to 20 kHz line rate range.

RATE2 specifies a 20 kHz to 25 kHz line rate range.

RATE3 specifies a 25 kHz to 35 kHz line rate range.

RATE4 specifies a 35 kHz to 50 kHz line rate range.

RATE5 specifies a 50 kHz to 65 kHz line rate range.

Examples

```
TRIGger:MAIn:VIDeo:SCAN RATE1  
specifies the line rate range for commercial broadcast television.
```

Related Commands

None

TRIGger:MAIn:VIDeo:SOURce

Sets or queries the source for the main video trigger. This is equivalent to selecting Source in the Video trigger menu.

Group

Trigger

Syntax

```
TRIGger:MAIn:VIDeo:SOUrce { CH<x> }
```

```
TRIGger:MAIn:VIDeo:SOUrce?
```

Arguments

CH<x> specifies one of the input channels (CH1 or CH2).

Examples

```
TRIGGER:MAIN:VIDEO:SOURCE CH1
```

selects channel 1 as the source for the main video trigger.

Related Commands

None

TRIGger:STATE? (Query Only)

Returns the current state of the oscilloscope triggering system.

Group

Trigger

Syntax

```
TRIGger:STATE?
```

Returns

REAdy indicates that all pretrigger information has been acquired and the oscilloscope is ready to accept a trigger.

PARTiAl indicates that the main trigger has occurred and the oscilloscope is waiting for the “runs after” delay to complete.

TRIGger indicates that the oscilloscope has seen a trigger and is acquiring the posttrigger information.

AUTO indicates that the oscilloscope is in auto mode and acquires data even in the absence of a trigger.

SAVE indicates that acquisition is stopped or that all channels are off.

ARMed indicates that the oscilloscope is acquiring pretrigger information. All triggers are ignored when TRIGger:STATE is ARMed.

Examples

TRIGGER:STATE?

might return READY, indicating that pretrigger data has been acquired and the oscilloscope is waiting for a trigger.

Related Commands

None

*TST? (Query Only)

(Self-Test) Tests the RS-232 interface and returns a 0.

Group

Miscellaneous Oscilloscope

Syntax

*TST?

Returns

None

Examples

None

Related Commands

None

UNLock (No Query Form)

Unlocks the front panel. This command is equivalent to LOCK NONE.

Group

Miscellaneous Oscilloscope

Syntax

UNLock ALL

Arguments

ALL specifies all front-panel buttons.

Examples

UNLock ALL

unlocks all front-panel buttons so they can be used.

Related Commands

LOCK

VERBose

Sets and queries the Verbose State that controls the length of keywords on query responses. Keywords can be both headers and arguments. This command does not affect IEEE Std 488.2–1987 Common Commands (those starting with an asterisk).

Group

Miscellaneous Oscilloscope

Syntax

VERBose { OFF | ON | <NR1> }

VERBose?

Arguments

ON or <NR1> \neq 0 sets the Verbose State true, which returns full-length keywords for applicable setting queries.

OFF or <NR1> = 0 sets the Verbose State false, which returns minimum-length keywords for applicable setting queries.

Examples

VERBOSE ON
sets the Verbose State true.

VERBOSE?
might return the value 1, showing that the Verbose State is true.

Related Commands

HEADer, *LRN?, SET?

*WAI (No Query Form)

*WAI (wait) prevents the instrument from executing further commands or queries until all pending operations finish. This command allows you to synchronize the operation of the instrument with your application program. Synchronization methods are described on page 3–9.

Group

Status and Error

Syntax

*WAI

Arguments

None

Examples

None

Related Commands

BUSY?, *OPC

WAVFrm? (Query Only)

Returns WFMPre? and CURVe? data for the waveform or waveforms as specified by the DATA:SOURce command. This command is equivalent to sending WFMPre?; CURVe?.

Group

Waveform

Syntax

WAVFrm?

Returns

See WFMPre? and CURVe? commands.

Examples

None

Related Commands

CURVe?, DATA:SOURce, WFMPre?

WFMPre? (Query Only)

Returns the waveform formatting data for the waveform or waveforms as specified by the DATA:SOURce command. Channel and math waveforms specified by the DATA:SOURce command must be displayed.

Group

Waveform

Syntax

WFMPre?

Returns

The format of the response is:

```
BYT_Nr <NR1>;BIT_Nr <NR1>;ENCdg { ASC | BIN };  
BN_Fmt { RI | RP };BYT_Or { LSB | MSB };
```

```
<wfm>:WFID <Qstring>;NR_PT <NR1>;PT_FMT { ENV | Y };  
XUNit <QString>;XINcr <NR3>;PT_Off <NR1>;YUNit  
<QString>;YMUlt <NR3>; YOff <NR3>;YZEro<NR3>[;<wfm>:  
WFID <Qstring>;NR_PT <NR1>;PT_FMT{ ENV | Y };  
XUNit<QString>;XINcr <NR3>;PT_Off <NR1>;YUNit <QString>;  
YMUlt <NR3>; YOff <NR3>;YZEro <NR3>...]
```

Examples

None

Related Commands

WAVFrm?

WFMPre:BIT_Nr

Returns the number of bits per binary waveform point for the waveform or waveforms as specified by the DATA:SOURce command. The WFMPre:BIT_Nr command is ignored on input.

Group

Waveform

Syntax

```
WFMPre:BIT_Nr <NR1>
```

```
WFMPre:BIT_Nr?
```

Arguments

<NR1> is either 8 or 16, and is equivalent to WFMPre:BYT_Nr * 8.

Examples

```
WFMPRE:BIT_NR?
```

might return 8, indicating that there are 8 bits per waveform point.

Related Commands

DATA:WIDth, WFMPre:BYT_Nr

WFMPre:BN_Fmt

Sets or queries the format of binary data for the waveform or waveforms specified by the DATA:SOURce command.

Group

Waveform

Syntax

```
WFMPre:BN_Fmt { RI | RP }
```

```
WFMPre:BN_Fmt?
```

Arguments

RI specifies signed integer data-point representation.

RP specifies positive integer data-point representation.

Examples

```
WFMPRE:BN_FMT RP
```

specifies that the binary waveform data are positive integer data-points.

```
WFMPRE:BN_FMT?
```

returns either RI or RP as the current waveform data format.

Related Commands

DATA:ENCdg, WFMPre:BYT_Or, WFMPre:ENCdg

WFMPre:BYT_Nr

Sets or queries the binary field data width for the first ordered waveform as specified by the DATA:SOURce command. This command is equivalent to the DATA:WIDTH command.

Group

Waveform

Syntax

WFMPre:BYT_Nr <NR1>

WFMPre:BYT_Nr?

Arguments

<NR1> is the number of bytes per point and can be 1 or 2.

Examples

WFMPRE:BYT_NR 2

specifies that there are 2 bytes per waveform data point.

Related Commands

DATA:WIDTH, WFMPre:BIT_Nr

WFMPre:BYT_Or

Selects which byte of binary waveform data is transmitted first during a waveform data transfer when DATA:WIDTH (or WFMPre:BYT_Nr) is set to 2.

Group

Waveform

Syntax

WFMPre:BYT_Or { LSB | MSB }

WFMPre:BYT_Or?

Arguments

LSB selects the least significant byte to be transmitted first.

MSB selects the most significant byte to be transmitted first.

Examples

WFMPRE:BYT_OR MSB

specifies that the most significant byte in the waveform data is transferred first.

WFMPRE:BYT_OR?

returns either MSB or LSB depending on which data byte is transferred first.

Related Commands

DATA:ENCdg, WFMPre:BN_Fmt, WFMPre:ENCdg

WFMPre:ENCdg

Sets or queries the type of encoding for waveform data transferred with the CURVe command.

Group

Waveform

Syntax

WFMPre:ENCdg { ASC | BIN }

WFMPre:ENCdg?

Arguments

ASC specifies ASCII curve data.

BIN specifies binary curve data.

Examples

WFMPRE:ENCDG ASC

specifies that the waveform data is in ASCII format.

WFMPRE:ENCDG?

might return BIN, indicating that the waveform data is in binary format.

Related Commands

DATA:ENCdg, WFMPre:BYT_Or, WFMPre:BN_Fmt

WFMPre:PT_Fmt (No Query Form)

Selects the point format of the waveform data for the first ordered waveform as specified by the DATA:SOURce command.

Group

Waveform

Syntax

```
WFMPre:PT_Fmt { ENV | Y }
```

Arguments

ENV specifies that the waveform is transmitted as maximum and minimum point pairs. Only y values are explicitly transmitted. Absolute coordinates are given by:

$$X_n = 0 + XINcr (n - PT_Off)$$

$$Y_{n_{max}} = YZero + YMult (y_{n_{max}} - YOYf)$$

$$Y_{n_{min}} = YZero + YMult (y_{n_{min}} - YOYf)$$

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:

$$X_n = 0 + XINcr (n - PT_Off)$$

$$Y_n = YZero + YMult (y_n - YOYf)$$

Examples

```
WFMPRE:PT ENV
```

sets the waveform data point format to enveloped.

Related Commands

None

WFMPre:PT_Off (No Query Form)

Specifies the trigger point within the waveform record for the reference waveform specified by the DATA:DESTINATION command.

Group

Waveform

Syntax

```
WFMPre:PT_Off <NR1>
```

Arguments

<NR1> = 0 to the record length (2500), and is the position of the trigger point relative to DATA:START (<nr1> can be negative).

Examples

```
WFMPRE:PT_OFF 1
```

specifies that the trigger point is the first point in the waveform record.

Related Commands

```
HORizontal:TRIGger:POsition
```

WFMPre:XINcr (No Query Form)

Specifies the horizontal sampling interval for the reference waveform specified by the DATA:DESTINATION command.

Group

Waveform

Syntax

```
WFMPre:XINcr <NR3>
```

Arguments

<NR3> is the sampling interval, in seconds per point.

Examples

None

Related Commands

None

WFMPre:YMUIt (No Query Form)

Specifies the vertical scale factor for the reference waveform specified by the DATA:DESTination command.

Group

Waveform

Syntax

WFMPre:YMUIt <NR3>

Arguments

<NR3> is the vertical scale factor, in YUNits (often volts) per division.

Examples

None

Related Commands

None

WFMPre:YOff (No Query Form)

Specifies the offset of the vertical component for the reference waveform specified by the DATA:DESTination command.

Group

Waveform

Syntax

WFMPre:YOff <NR3>

Arguments

<NR3> is the vertical offset in digitizing levels.

Examples

None

Related Commands

None

WFMPre:YZero (No Query Form)

Specifies the offset for the reference waveform specified by the DATA:DESTINATION command.

Group

Waveform

Syntax

WFMPre:YZero <NR3>

Arguments

<NR3> is of the offset, in YUNits (often volts).

Examples

None

Related Commands

None

Table 2–31 lists additional WFMPre commands that are included for compatibility purposes.

NOTE. *These commands do not support a query form, and all information is ignored.*

Table 2–31: Additional WFMPre Commands

Command	Argument	Description
WFMPre:NR_PT	<NR1>	Number of waveform points
WFMPre:WFIId	<QString>	Waveform identifier
WFMPre:XUNit	<QString>	Horizontal units
WFMPre:XMU1t	<NR3>	Horizontal (X-axis) scale factor
WFMPre:XOff	<NR3>	Horizontal (X-axis) offset
WFMPre:XZErO	<NR3>	Horizontal (X-axis) origin offset
WFMPre:YUNit	<QString>	Vertical units
WFMPre:ZMU1t	<NR3>	Z-axis scale factor
WFMPre:ZOff	<NR3>	Z-axis offset
WFMPre:ZUNit	<QString>	Z-axis units
WFMPre:ZZErO	<NR3>	Z-axis origin offset

NOTE. When returning `WFMPRE:<wfm>` information from the instrument, `<wfm>` specifies the waveform source (`CH<x>`, `MATH`, `STO[1–10]`, or `REF<x>`). The source must also be set using the `DATA:SOURce` command. When sending `WFMPRE:<wfm>` information to the instrument, the `<wfm>` specification is ignored and the reference location specified by `DATA:DESTination` is used instead.

WFMPre:<wfm>? (Query Only)

Returns the waveform formatting data for the waveform specified by the `DATA:SOURce` command. Channel and math waveforms must be displayed before they can be queried. Querying an invalid stored waveform generates an execution error.

Group

Waveform

Syntax

`WFMPre:<wfm>?`

Returns

The format of the response is:

```
<wfm>:WFID <Qstring>;NR_PT <NR1>;PT_FMT { ENV | Y };
XUnit <QString>;XINcr <NR3>;PT_Off <NR1>;YUnit
<QString>;YMult <NR3>;YOff <NR3>;YZero <NR3>[;<wfm>
:WFID <Qstring>;NR_PT <NR1>;PT_FMT { ENV | Y };
XUnit <QString>;XINcr <NR3>;PT_Off <NR1>;YUnit
<QString>;YMult <NR3>;YOff <NR3>;YZero <NR3>...]
```

Examples

None

Related Commands

None

WFMPre:<wfm>:NR_Pt

Sets or queries the number of points that are in the transmitted waveform record. This value is ignored on input.

Group

Waveform

Syntax

WFMPre:<wfm>:NR_Pt <NR1>

WFMPre:<wfm>:NR_Pt?

Arguments

<NR1> is the number of data points. If DATA:WIDTH is 2, then there are twice as many bytes.

<NR1> = 0 means that the waveform record is of an unspecified length.

Examples

WFMPRE:CH1:NR_Pt?

might return 2500 as the number of data points in the waveform record transferred from channel 1.

Related Commands

DATA:DESTination

WFMPre:<wfm>:PT_Fmt

Selects the data point format for the waveform selected by the DATA:SOURce command. On input <wfm> always defaults to the reference location specified by DATA:DESTination regardless of what is sent.

Group

Waveform

Syntax

WFMPre:<wfm>:PT_Fmt { ENV | Y }

WFMPre:<wfm>:PT_Fmt?

Arguments

ENV specifies that the waveform is transmitted as maximum and minimum point pairs. Only y values are explicitly transmitted. Absolute coordinates are given by:

$$X_n = 0 + XINcr (n - PT_Off)$$

$$Y_{n_{max}} = YZEro + YMUlt (y_{n_{max}} - YOFf)$$

$$Y_{n_{min}} = YZEro + YMUlt (y_{n_{min}} - YOFf)$$

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:

$$X_n = 0 + XINcr (n - PT_Off)$$

$$Y_n = YZEro + YMUlt (y_n - YOFf)$$

Examples

WFMPRE:MATH:PT_FMT?

might return ENV, indicating that the MATH waveform data format is enveloped.

Related Commands

DATA:DESTination

WFMPre:<wfm>:PT_Off

Returns the trigger point within the waveform record. On input <wfm> always defaults to the reference location specified by DATA:DESTination regardless of what is sent.

Group

Waveform

Syntax

WFMPre:<wfm>:PT_Off <NR1>

WFMPre:<wfm>:PT_Off?

Arguments

<NR1> = 0 to the record length (2500), and is the position of the trigger point relative to DATA:STARt when queried.

Examples

WFMPRE:CH1:PT_OFF?

returns 0 indicating the trigger position within the waveform record.

Related Commands

None

WFMPre:<wfm>:WFId? (Query Only)

Returns information about the waveform such as input coupling, volts/division, time/division, acquisition mode, and record length.

The WFMPre:<wfm>:WFId command is ignored on input.

Group

Waveform

Syntax

WFMPre:<wfm>:WFId?

Returns

None

Examples

WFMPre:CH1:WFId?

might return :WFMPRE:CH1:WFID "Ch1, DC coupling, 1.0E-1 Volts/div, 5.0E-5 sec/div, 2500 points, Sample mode"

Related Commands

None

WFMPre:<wfm>:XINcr

Sets or queries the horizontal sampling interval of the specified waveform. The command and query form ignores the <wfm> parameter; instead, the instrument uses the reference location specified by DATA:DESTination command.

Group

Waveform

Syntax

WFMPre:<wfm>:XINcr <NR3>

WFMPre:<wfm>:XINcr?

Arguments

<NR3> is the sampling interval.

Examples

WFMPre:CH1:XINcr
might return :WFMPRE:CH1:XINCR 2.0E-6

Related Commands

None

WFMPre:<wfm>:XUNit? (Query Only)

Returns the horizontal (X-axis) units of the waveform data at the time of acquisition.

The WFMPre:<wfm>:XUNit command is ignored on input.

Group

Waveform

Syntax

WFMPre:<wfm>:XUNit?

Returns

None

Examples

WFMPRE:CH1:XUNIT?

returns "s", indicating that the horizontal units for channel 1 are seconds.

Related Commands

None

WFMPre:<wfm>:YMUIt

Sets or queries the vertical scale factor, in YUNit(s) per digitizing level. The command and query form ignores the <wfm> parameter; instead, the instrument uses the reference location specified by DATA:DESTination command.

Group

Waveform

Syntax

WFMPre:<wfm>:YMUIt <NR3>

WFMPre:<wfm>:YMUIt?

Arguments

<NR3> is the scale factor, in YUNits (often volts) per digitizing level.

Examples

WFMPre:CH1:YMUIt?

might return :WFMPRE:CH1:YMULT 4.0E-2

Related Commands

None

WFMPre:<wfm>:YOFF

Sets or queries the vertical position of the waveform. On input, <wfm> always defaults to the reference location specified by DATA:DESTINATION regardless of what is sent. The command and query form ignores the <wfm> parameter; instead, the instrument uses the reference location specified by DATA:DESTINATION command.

Group

Waveform

Syntax

WFMPre:<wfm>:YOFF <NR3>

WFMPre:<wfm>:YOFF?

Arguments

<NR3> is the position in digitizing levels.

Examples

WFMPre:CH1:YOFF?

might return :WFMPRE:CH1:YOFF 2.4E1

Related Commands

None

WFMPre:<wfm>:YUNit? (Query Only)

Returns the vertical (Y-axis) units of the waveform data at the time of acquisition.

The WFMPre:<wfm>:YUNit command is ignored on input.

Group

Waveform

Syntax

WFMPre:<wfm>:YUNit?

Returns

None

Examples

WFMPRE:CH2:YUNIT?

might return "A", meaning that the units for the vertical component of the channel 2 waveform data are amperes.

Related Commands

None

WFMPre:<wfm>:YZEro

Sets or queries the vertical (Y-axis) offset voltage. On input, <wfm> always defaults to the reference location specified by DATA:DESTINATION regardless of what is sent. The command and query form ignores the <wfm> parameter; instead, the instrument uses the reference location specified by DATA:DESTINATION command.

Group

Waveform

Syntax

WFMPre:<wfm>:YZEro <NR3>

WFMPre:<wfm>:YZEro?

Arguments

<NR3> is the offset in YUNits (often volts).

Examples

WFMPre:CH1:YZEro?

always returns :WFMPRE:CH1:YZERO 0.0E0

Related Commands

None

ZOOM:STATE

Included for compatibility purposes only.

Group

Zoom

Syntax

ZOOM:STATE { OFF | ON | <NR1> }

ZOOM:STATE?

Arguments

None

Examples

None

Related Commands

None

ZOOM:VERTical:POSition

Sets or queries the vertical position of waveforms.

NOTE. ZOOM commands affect only REF and MATH waveforms.

Group

Zoom

Syntax

ZOOM:VERTical:POSition <NR3>

ZOOM:VERTical:POSition?

Arguments

<NR3> is the vertical position, in divisions.

Examples

ZOOM:VERTICAL:POSITION?

might return :ZOOM:VERTICAL:POSITION 0

Related Commands

None

ZOOM:VERTICAL:SCALE

Sets or queries the vertical expansion and compression factor.

***NOTE.** ZOOM commands affect only REF and MATH waveforms.*

Group

Zoom

Syntax

ZOOM:VERTICAL:SCALE <NR3>

ZOOM:VERTICAL:SCALE?

Arguments

<NR3> is the amount of vertical expansion or compression.

Examples

ZOOM:VERTICAL:SCALE?

might return :ZOOM:VERTICAL:SCALE 1.0E0

Related Commands

None



Status and Events

Status and Events

The TekScope instrument reports status and events over the RS-232 interface. The status and event reporting system emulates GPIB behavior with the RS-232 interface to provide compatibility with existing host software.

The instrument status reporting system consists of five 8-bit registers and two queues. This section describes these registers and components, and explains how the event handling system operates.

Registers

The registers in the event reporting system fall into two functional groups:

- The Standard Event Status Register (SESR) and the Status Byte Register (SBR) contain information about the status of the instrument. These registers are the Status Registers.
- The Device Event Status Enable Register (DESER), the Event Status Enable Register (ESER), and the Service Request Enable Register (SRER) determine whether selected types of events are reported to the Status Registers and the Event Queue. These three registers are the Enable Registers.

Status Registers

The Standard Event Status Register (SESR) and the Status Byte Register (SBR) record certain types of events that may occur while the instrument 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 service request. When an event of a given type occurs, the instrument 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 the Enable Registers section on page 3–4.) Reading the status registers tells you what types of events have occurred.

The Standard Event Status Register (SESR). The SESR, shown in Figure 3–1, records eight types of events that can occur within the instrument. Use *ESR? to read the SESR register. Reading the register clears the bits of the register so that the register can accumulate information about new events. Table 3–1 shows SESR bit functions.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Figure 3–1: The Standard Event Status Register (SESR)

Table 3–1: SESR Bit Functions

Bit	Function
7 (MSB)	PON (Power On). Shows that the instrument was powered on.
6	URQ (User Request). Not used.
5	CME (Command Error). Shows that an error occurred while the instrument was parsing a command or query. Command error messages are listed in Table 3–4 on page 3–14.
4	EXE (Execution Error). Shows that an error occurred while the instrument was executing a command or query. Execution error messages are listed in Table 3–5 on page 3–15.
3	DDE (Device Error). Shows that a device error occurred. Device error messages are listed in Table 3–6 on page 3–18.
2	QYE (Query Error). Shows that data in the Output Queue was lost.
1	RQC (Request Control). Not used.
0 (LSB)	OPC (Operation Complete). Shows that the operation is complete. This bit is set when all pending operations complete following a *OPC command.

The Status Byte Register (SBR). The SBR, shown in Figure 3–2, records whether output is available in the Output Queue, whether the instrument requests service, and whether the SESR has recorded any events.

Use *STB? 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 the *STB? query to obtain the SBR, bit 6 is the MSS bit. Reading the SBR does not clear the bits. Table 3–2 shows the SBR bit functions.

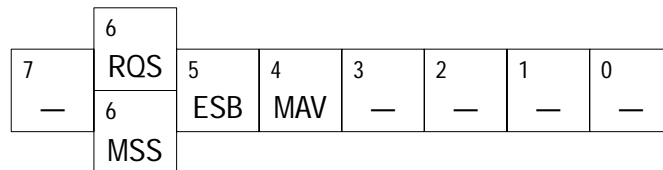


Figure 3–2: The Status Byte Register (SBR)

Table 3–2: SBR Bit Functions

Bit	Function
7 (MSB)	Not used.
6	RQS (Request Service) Shows that the instrument requests service from the controller.
6	MSS (Master Status Summary), obtained from *STB?. Summarizes the ESB and MAV bits in the SBR.
5	ESB (Event Status Bit). Shows that status is enabled and present in the SESR.
4	MAV (Message Available). Shows that output is available in the Output Queue.
3 – 0	Not used.

Enable Registers

The 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 its 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.

The bits in the Enable Registers are set using various commands. The Enable Registers and the commands used to set them are described below.

The Device Event Status Enable Register (DESER). The DESER, shown in Figure 3–3, 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, as described earlier.

Use the DESE command to enable and disable the bits in the DESER. Use the DESE? query to read the DESER.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Figure 3–3: The Device Event Status Enable Register (DESER)

The Event Status Enable Register (ESER). The ESER, shown in Figure 3–4, 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, and use the *ESE? query to read it.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Figure 3–4: The Event Status Enable Register (ESER)

The Service Request Enable Register (SRER). The SRER, shown in Figure 3–5, controls which bits in the SBR are summarized by the Master Status Summary (MSS) bit.

Use the *SRE command to set the SRER. Use the *SRE? query to read it. The RQS bit remains set to one until the MSS bit changes back to a zero.

7	6	5	4	3	2	1	0
—	—	ESB	MAV	—	—	—	—

Figure 3–5: The Service Request Enable Register (SRER)

The Enable Registers and the *PSC Command

The *PSC command controls the contents of the Enable Registers 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.

Queues

The instrument status and event reporting system contains two queues: the Output Queue and the Event Queue.

The Output Queue

The Output Queue stores query responses waiting to be output.

The Event Queue

The Event Queue stores detailed information on up to 20 events. If more than 20 events stack up in the Event Queue, the 20th event is replaced by event code 350, “Too many events.”

Read the Event Queue with EVENT? (which returns only the event number), with EVMsg? (which returns the event number and a text description of the event), or with ALLEV? (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 *ESR? to read the summary of the event from the SESR. This makes the events summarized by *ESR? available to EVENT? and EVMSG?, and empties the SESR.

Reading the SESR erases any events that were summarized by previous *ESR? reads but not read from the Event Queue. Events that follow an *ESR? read are put in the Event Queue, but are not available until *ESR? is used again.

Event Handling Sequence

Figure 3–6 shows how to use the status and event handling system. In the explanation that follows, numbers in parentheses refer to numbers in Figure 3–6.

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 (GPIB only) is generated (7).

Status and Events

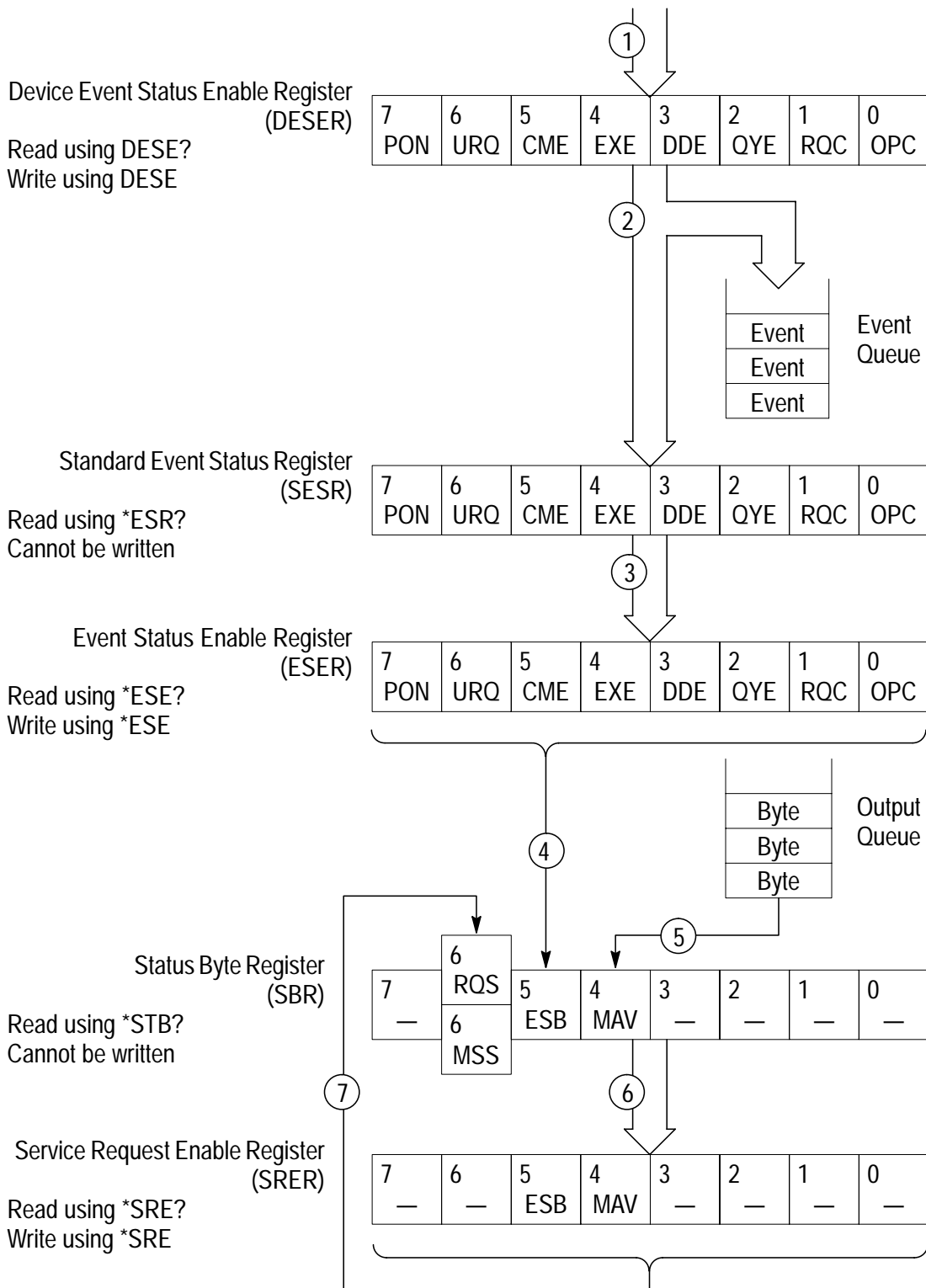


Figure 3–6: Status and Event Handling Process

Synchronization Methods

Although most commands are completed almost immediately after being received by the instrument, some commands start a process that requires more time. For example, once a single-sequence acquisition is started, it may be a few seconds before the acquisition is complete. Rather than remain idle while the operation is in process, the instrument continues processing other commands. This means that some operations are not completed in the order that they were sent.

There may be times when the result of an operation is dependent on the result of an earlier one, and you must be assured that the first operation has completed before processing the next one. The status and event reporting system provides ways to do this.

For example, a typical application would be to acquire a single-sequence waveform, and then take a measurement on the acquired waveform. You could use the following command sequence:

```
/** Set up single-sequence acquisition **/  
SELECT:CH1 ON  
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 on acquired data **/  
MEASUREMENT:IMMED:VALUE?
```

The acquisition of the waveform requires extended processing time and may not complete before the amplitude measurement is taken. (See Figure 3–7.) This will result in an incorrect amplitude value.

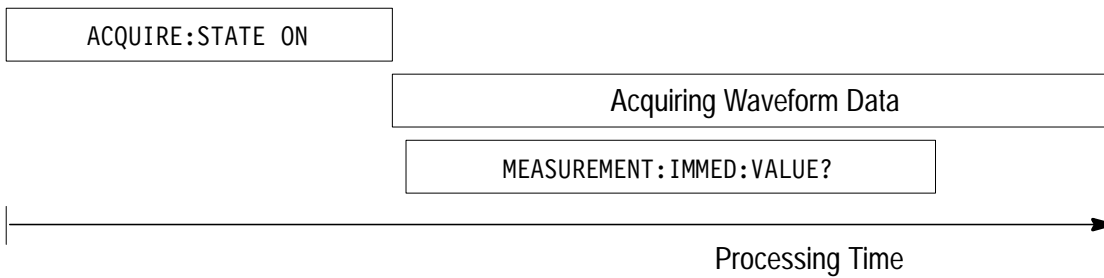


Figure 3-7: Command Processing Without Using Synchronization

The acquisition of the waveform must be completed before the measurement can be taken on the acquired data. This is achieved by synchronizing the program so that the measurement command is not processed by the instrument until the acquisition is complete. Figure 3-8 shows the desired processing sequence.

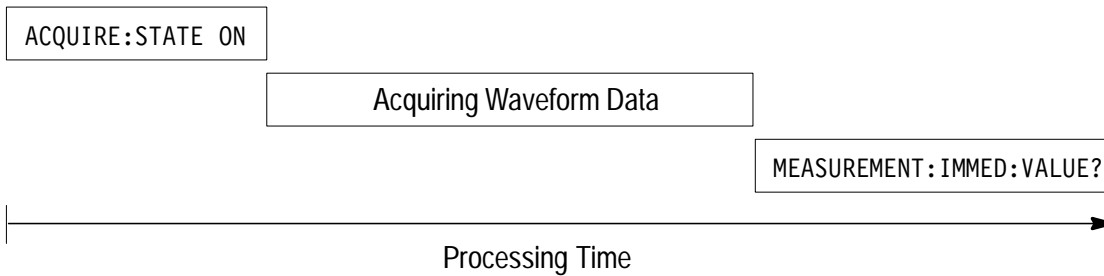


Figure 3-8: Processing Sequence With Synchronization

Four commands can be used to synchronize the operation of the instrument with your application program: *WAI, BUSY?, *OPC, and *OPC?.

Using the *WAI Command

You can force commands to execute sequentially by using the *WAI command. This command forces completion of the previous commands before processing new ones.

The same command sequence using the *WAI command for synchronization follows:

```
/* Set up single-sequence acquisition */
SELECT:CH1 ON
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 on acquired data */
MEASUREMENT:IMMED:VALUE?
```

Though *WAI is one of the easiest ways to achieve synchronization, it is also the most costly. The processing time of the instrument is slowed, since it is processing a single command at a time. This time could be spent doing other tasks.

The controller can continue to write commands to the input buffer, but the commands are not processed by the instrument until all operations in process are complete. If the input buffer becomes full, the controller will be unable to write any more commands to the buffer and will result in a time out.

Using the BUSY Query

BUSY? allows you to find out whether the instrument is busy processing a command that has an extended processing time, such as single-sequence acquisition.

The same command sequence using `BUSY?` for synchronization follows:

```
/* Set up single-sequence acquisition */
SELECT:CH1 ON
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* Wait until the acquisition is complete before taking the
measurement */
While BUSY? keep looping
/* Take amplitude measurement on acquired data */
MEASUREMENT:IMMED:VALUE?
```

This sequence lets you create your own wait loop rather than using the `*WAI` command. An advantage to using `BUSY?` is that you eliminate the possibility of a time out caused by writing too many commands to the input buffer. The controller is still tied up, though, and the repeated `BUSY?` results in more serial data 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 can use this command in conjunction with an RS-232 interrupt handler to achieve synchronization.

Using the `*OPC` Query

`*OPC?` places a 1 in the Output Queue once an operation is complete.

The same command sequence using *OPC? for synchronization follows:

```

/* Set up single-sequence acquisition */
SELECT:CH1 ON
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 on acquired data */
MEASUREMENT:IMMED:VALUE?

```

Using *OPC? synchronization is the simplest approach. It requires no status handling or loops. However, you must set the time out for RS-232 operations in the controller for longer than the acquisition operation.

Messages

Tables 3–3 through 3–9 list all the programming interface messages the instrument generates in response to commands and queries.

For most messages, a secondary message from the instrument gives more 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 that follow, the associated SESR bit is specified in the table title, with exceptions noted with the error message text.

Table 3–3 shows the messages when the system has no events or status to report. These have no associated SESR bit.

Table 3–3: No Event Messages

Code	Message
0	No events to report – queue empty
1	No events to report – new events pending *ESR?

Table 3–4 shows the error messages generated by improper command syntax. Check that the command is properly formed and that it follows the rules in the *Command Syntax* section starting on page 2–1.

Table 3–4: Command Error Messages – CME Bit 5

Code	Message
100	Command error
102	Syntax error
103	Invalid separator
104	Data type error
105	GET not allowed
108	Parameter not allowed
110	Command header error
111	Header separator error
112	Program mnemonic too long

Table 3–4: Command Error Messages – CME Bit 5 (Cont.)

Code	Message
113	Undefined header
161	Invalid block data (indefinite length blocks are not allowed over the RS-232)

Table 3–5 lists the errors that are detected during execution of a command. In these error messages, you should read “macro” as “alias.”

Table 3–5: Execution Error Messages – EXE Bit 4

Code	Message
200	Execution error
201	Invalid while in local
210	Trigger error
211	Trigger ignored
212	Arm ignored
220	Parameter error
221	Settings conflict
222	Data out of range
223	Too much data
224	Illegal parameter value
230	Data corrupt or stale
240	Hardware error
241	Hardware missing
242	Hardware configuration error
243	Hardware I/O device error
260	Expression error

Table 3–5: Execution Error Messages – EXE Bit 4 (Cont.)

Code	Message
261	Math error in expression
2200	Measurement error, Measurement system error
2201	Measurement error, Zero period
2202	Measurement error, No period found
2203	Measurement error, No period, second waveform
2204	Measurement error, Low signal amplitude
2205	Measurement error, Low amplitude, second waveform
2206	Measurement error, Invalid gate
2207	Measurement error, Measurement overflow
2208	Measurement error, Waveform does not cross Mid Ref
2209	Measurement error, No second Mid Ref crossing
2210	Measurement error, No Mid Ref crossing, second waveform
2211	Measurement error, No backwards Mid Ref crossing
2212	Measurement error, No negative crossing
2213	Measurement error, No positive crossing
2214	Measurement error, No crossing
2215	Measurement error, No crossing, second waveform
2216	Measurement error, No crossing, target waveform
2217	Measurement error, Constant waveform
2218	Measurement error, Unused
2219	Measurement error, No valid edge – No arm sample
2220	Measurement error, No valid edge – No arm cross
2221	Measurement error, No valid edge – No trigger cross
2222	Measurement error, No valid edge – No second cross
2223	Measurement error, waveform mismatch

Table 3–5: Execution Error Messages – EXE Bit 4 (Cont.)

Code	Message
2224	Measurement error, WAIT calculating
2225	Measurement error, No waveform to measure
2226	Null Waveform
2227	Positive and Negative Clipping
2228	Measurement error, Positive Clipping
2229	Measurement error, Negative Clipping
2230	Measurement error, High Ref < Low Ref
2235	Math error, Invalid math description
2240	Invalid password
2241	Waveform request is invalid
2242	Data start and stop > record length
2243	Waveform requested is not a data source
2244	Waveform requested is not turned on
2245	Saveref error, Selected channel is turned off
2246	Saveref error, Selected channel data invalid
2248	Saveref error, Source reference data invalid
2260	Calibration error
2270	Alias error
2271	Alias syntax error
2272	Alias execution error
2273	Illegal alias label
2274	Alias parameter error
2275	Alias definition too long
2276	Alias expansion error
2277	Alias redefinition not allowed

Table 3–5: Execution Error Messages – EXE Bit 4 (Cont.)

Code	Message
2278	Alias header not found
2279	Alias label too long
2280	Alias table full
2285	Tek Secure® Pass
2286	Tek Secure® Fail
2301	Cursor error, Off screen

Table 3–6 lists the device errors that can occur during instrument operation. These errors may indicate that the oscilloscope needs repair.

Table 3–6: Device Error Messages – DDE Bit 3

Code	Message
300	Device-specific error
310	System error
311	Memory error
312	PUD memory lost
313	Calibration memory lost
314	Save/recall memory lost
315	Configuration memory lost
350	Queue overflow (does not set DDE bit)
361	Parity error in program message (check parity)
362	Framing error in program message (check baud rate)
363	Input buffer overrun (check flagging)

Table 3–7 lists the system event messages. These messages are generated whenever certain system conditions occur.

Table 3–7: System Event Messages

Code	Message
400	Query event
401	Power on (PON bit 7 set)
402	Operation complete (OPC bit 0 set)
403	User request (URQ bit 6 set)
404	Power fail (DDE bit 3 set)
405	Request control
410	Query INTERRUPTED (QYE bit 2 set)
420	Query UNTERMINATED (QYE bit 2 set)
430	Query DEADLOCKED (QYE bit 2 set)
440	Query UNTERMINATED after indefinite response (QYE bit 2 set)

Table 3–8 lists warning messages that do not interrupt the flow of command execution. These notify you that you may get unexpected results.

Table 3–8: Execution Warning Messages – EXE Bit 4

Code	Message
500	Execution warning
510	String data too long, truncated
525	Parameter underrange
526	Parameter overrange
527	Parameter rounded
528	Parameter out of range

Table 3–8: Execution Warning Messages – EXE Bit 4 (Cont.)

Code	Message
530	Data stop > stop, Values swapped internally
531	Data stop > record length, Curve truncated
532	Curve data too long, Curve truncated
540	Measurement warning
541	Measurement warning, Low signal amplitude
542	Measurement warning, Unstable histogram
543	Measurement warning, Low resolution
544	Measurement warning, Uncertain edge
545	Measurement warning, Invalid in minmax
546	Measurement warning, Need 3 edges
547	Measurement warning, Clipping positive/negative
548	Measurement warning, Clipping positive
549	Measurement warning, Clipping negative

Table 3–9 shows internal errors that indicate an internal fault in the instrument.

Table 3–9: Internal Warning Messages

Code	Message
600	Internal warning



Examples

Programming Examples

The example programs illustrate methods you can use to control the TekScope instrument from the RS-232 interface. The diskette that comes with this manual contains listings for these programs.

The programs run on a PC compatible system equipped with an RS-232 (COM2) serial port.

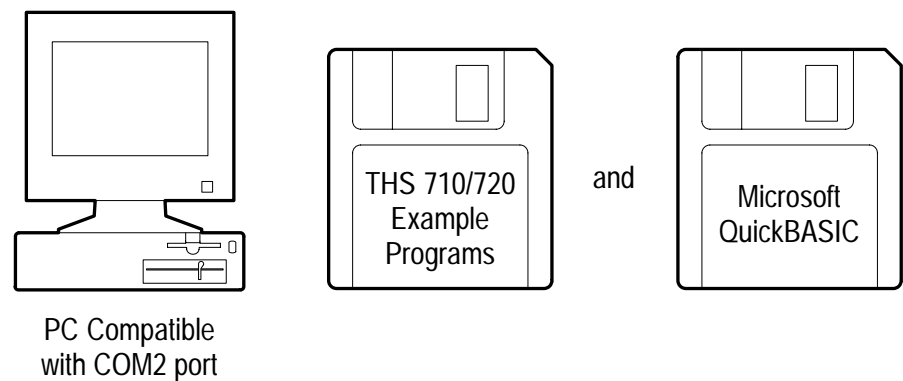


Figure 4-1: Equipment Needed to Run the RS-232 Example Programs

The example RS-232 software includes the following functions:

- MEAS: measures a parameter of an oscilloscope waveform.
- COMM: shows communication between the personal computer and the oscilloscope.
- GETWFM: reads a waveform from the instrument and stores its time and voltage values in a file.
- TL: is a talker-listener program.
- DMM: reads a DMM measurement from the instrument.

Compiling the Example Programs

The example programs diskette contains programs written in Microsoft QuickBASIC 4.5.

Executable versions of the programs are in the PROGRAMS directory. Source versions are in the SOURCES directory. Within this directory, the QuickBASIC programs are in the Q-BASIC subdirectory.

A README file in each directory explains how to build executable code from the source files provided.

***NOTE.** The programs you compile in the Sources directory may require extra steps or changes to get them to work with your system.*

To make an executable for any of the following files, perform the following steps:

1. Install QuickBASIC.
2. Connect the oscilloscope to the COM2 port of the personal computer. Set the oscilloscope RS-232 parameters to the default values.
3. Copy the files from the examples diskette to your hard disk. You might also create a special directory to store them. For example, if the current drive is hard disk C, you want to store the examples in drive C, and the examples diskette is in drive B, type:

```
mkdir examples
cd examples
copy b:\rs232\q-basic\*.* .
```

4. Compile the programs using the following commands:

```
bc /o <file>.bas;
```

```
bc /o /v rs232io.bas;
```

where <file> is one of the example program names.

To compile MEAS.BAS, type: `bc /o meas.bas;`

To compile COMM.BAS, type: `bc /o comm.bas;`

To compile GETWFM.BAS, type: `bc /o getwfm.bas;`

To compile TL.BAS, type: `bc /o tl.bas;`

To compile DMM.BAS, type: `bc /o dmm.bas`

5. Link the compiled program with the rs232io.obj module to create the executable program (file.EXE) using the following command:

```
link <file>.obj+rs232io.obj;
```

where <file> is one of the above program names.

To link MEAS.OBJ, type: `link meas.obj+rs232io.obj;`

To link COMM.OBJ, type: `link comm.obj+rs232io.obj;`

To link GETWFM.OBJ, type: `link getwfm.obj+rs232io.obj;`

To link TL.OBJ, type: `link tl.obj+rs232io.obj;`

To link DMM.OBJ, type: `link dmm.obj+rs232io.obj;`

The rs232IO.BAS file is a collection of input/output routines used by the other programs and is included for proper file compilation.

6. Run the program by typing the program name.

To run MEAS, type: `meas`

To run COMM, type: `comm`

To run GETWFM, type: `getwfm`

To run TL, type: `tl`

To run DMM, type: `dmm`



Appendices

Appendix A: ASCII Code Chart

	0	1	2	3	4	5	6	7
0	⁰ NUL _{0 0}	²⁰ DLE _{10 16}	⁴⁰ SP _{20 32}	⁶⁰ 0 _{30 48}	¹⁰⁰ @ _{40 64}	¹²⁰ P _{50 80}	¹⁴⁰ , _{60 96}	¹⁶⁰ p _{70 112}
1	¹ SOH _{1 1}	²¹ DC1 _{11 17}	⁴¹ ! _{21 33}	⁶¹ 1 _{31 49}	¹⁰¹ A _{41 65}	¹²¹ Q _{51 81}	¹⁴¹ a _{61 97}	¹⁶¹ q _{71 113}
2	² STX _{2 2}	²² DC2 _{12 18}	⁴² " _{22 34}	⁶² 2 _{32 50}	¹⁰² B _{42 66}	¹²² R _{52 82}	¹⁴² b _{62 98}	¹⁶² r _{72 114}
3	³ ETX _{3 3}	²³ DC3 _{13 19}	⁴³ # _{23 35}	⁶³ 3 _{33 51}	¹⁰³ C _{43 67}	¹²³ S _{53 83}	¹⁴³ c _{63 99}	¹⁶³ s _{73 115}
4	⁴ EOT _{4 4}	²⁴ DC4 _{14 20}	⁴⁴ \$ _{24 36}	⁶⁴ 4 _{34 52}	¹⁰⁴ D _{44 68}	¹²⁴ T _{54 84}	¹⁴⁴ d _{64 100}	¹⁶⁴ t _{74 116}
5	⁵ ENQ _{5 5}	²⁵ NAK _{15 21}	⁴⁵ % _{25 37}	⁶⁵ 5 _{35 53}	¹⁰⁵ E _{55 69}	¹²⁵ U _{55 85}	¹⁴⁵ e _{65 101}	¹⁶⁵ u _{75 117}
6	⁶ ACK _{6 6}	²⁶ SYN _{16 22}	⁴⁶ & _{26 38}	⁶⁶ 6 _{36 54}	¹⁰⁶ F _{46 70}	¹²⁶ V _{56 86}	¹⁴⁶ f _{66 102}	¹⁶⁶ v _{76 118}
7	⁷ BEL _{7 7}	²⁷ ETB _{17 23}	⁴⁷ ' _{27 39}	⁶⁷ 7 _{37 55}	¹⁰⁷ G _{47 71}	¹²⁷ W _{57 87}	¹⁴⁷ g _{67 103}	¹⁶⁷ w _{77 119}
8	¹⁰ BS _{8 8}	³⁰ CAN _{18 24}	⁵⁰ (_{28 40}	⁷⁰ 8 _{38 56}	¹¹⁰ H _{48 72}	¹³⁰ X _{58 88}	¹⁵⁰ h _{68 104}	¹⁷⁰ x _{78 120}
9	¹¹ HT _{9 9}	³¹ EM _{19 25}	⁵¹) _{29 41}	⁷¹ 9 _{39 57}	¹¹¹ I _{49 73}	¹³¹ Y _{59 89}	¹⁵¹ i _{69 105}	¹⁷¹ y _{79 121}

Continued on next page

Appendix A: ASCII Code Chart

	0	1	2	3	4	5	6	7
A	¹² LF A 10	³² SUB 1A 26	⁵² * 2A 42	⁷² : 3A 58	¹¹² J 4A 74	¹³² Z 5A 90	¹⁵² j 6A 106	¹⁷² z 7A 122
B	¹³ VT B 11	³³ ESC 1B 27	⁵³ + 2B 43	⁷³ ; 3B 59	¹¹³ K 4B 75	¹³³ [5B 91	¹⁵³ k 6B 107	¹⁷³ { 7B 123
C	¹⁴ FF C 12	³⁴ FS 1C 28	⁵⁴ , 2C 44	⁷⁴ < 3C 60	¹¹⁴ L 4C 76	¹³⁴ \ 5C 92	¹⁵⁴ l 6C 108	¹⁷⁴ 7C 124
D	¹⁵ CR D 13	³⁵ GS 1D 29	⁵⁵ - 2D 45	⁷⁵ = 3D 61	¹¹⁵ M 4D 77	¹³⁵] 5D 93	¹⁵⁵ m 6D 109	¹⁷⁵ } 7D 125
E	¹⁶ SO E 14	³⁶ RS 1E 30	⁵⁶ . 2E 46	⁷⁶ > 3E 62	¹¹⁶ N 4E 78	¹³⁶ ^ 5E 94	¹⁵⁶ n 6E 110	¹⁷⁶ ~ 7E 126
F	¹⁷ SI F 15	³⁷ US 1F 31	⁵⁷ / 2F 47	⁷⁷ ? 3F 63	¹¹⁷ O 4F 79	¹³⁷ - 5F 95	¹⁵⁷ o 6F 111	¹⁷⁷ DEL (RUBOUT) 7F 127

KEY

octal

25

NAK

ASCII character

hex

15

21

decimal

Appendix B: Reserved Words

A list of the reserved words of the TekScope instrument follows. Do not use these words for aliases.

*CAL	BEEP
*CLS	BIT_Nr
*DDT	BMENU1
*ESE	BMENU2
*ESR	BMENU3
*IDN	BMENU4
*IST	BMENU5
*LRN	BN_Fmt
*OPC	BUSY
*PSC	BYT_Nr
*PUD	BYT_Or
*RCL	CALibrate
*RST	CATalog
*SAV	CH1
*SRE	CH1PROBE
*STB	CH2
*TRG	CH2PROBE
*TST	CLEARMenu
*WAI	CONTINUE
ABSolute	CONTinuity
ACQuire	CONTRast
ACQUISition	CONTR01
ACV	COUPling
ADJust	CPU
ALias	CURSor
ALL	CURS0R1
ALLEV	CURS0R2
ALWAYS	CURVe
AUTORange	DATa
AUTOSet	DCD
AVG	DCV
BACKlight	DEFINE
BANdwidth	DELay
BAUD	DELEte

Appendix B: Reserved Words

DELTA	HOLD
DESE	HOLDoff
DESTination	HORizontal
DEVelop	HORZMENU
DIAG	HORZPOS
DINK	HORZSCALE
DIODE	HPOS1
DISplay	HPOS2
DMM	ID
EDGE	IMMed
ENCdg	INTENSITY
ERASEFactory	INTERna1
ERRLOG	INVert
EVEnt	KEYboard
EVMsg	LAYout
EVQty	LEVe1
EXECute	LOCK
EXT	LOG
EXT10	LOOP
FACTory	LOW
FALL	MAG
FIELD1	MAIn
FIELD2	MATH
FITtoscreen	MATH1
FLAg	MAX
FORCE	MAXMin
FORMat	MEAS1
FREQuency	MEAS2
FPAnel	MEAS3
FUNCTion	MEAS4
GATING	MEASMENU
GPROCKER	MEASUrement
GRAticule	METer
HARDCopy	METHod
HARDFlagging	MID
HBArs	MIN
HDELta	MODE
HDR	NEWMax
HEADer	NEWpass
HEALTH	NOISYgain
HIGH	NR_Pt

NUMACq	SAVE
NUMAVg	SBITS
NUMEnv	SCALE
NUMERRors	SCANrate
OFFSet	SCOpe
OHMS	SECdiv
OPTion	SElect
OVERAll	SET
PACE	SETLevel
PAIRed	SETUp
PARity	SLOpe
PASSWord	SOFTFlagging
PERCent	SOURce
PERSistence	SOURCE1
POLarity	SRSETUP
PORT	STANDby
POSition	START
POSITION1	STATE
POSITION2	STATUS
POWER	STOP
PRESet	STOPAfter
PREss	STOPBits
PRObe	STYLE
PT_Fmt	TARget
PT_Off	TEKSecure
PULSE	TERMinator
RANGE	TEXT
RECA11	TIME
RECOrdlength	TOGGLE
REFA	TPDiv
REFB	TRANsmit
REFLevel	TRIGger
REJect	TRIGLEVEL
REM	TRIGMENU
RUNSTop	TRIGT
RESet	TYPE
RESUlT	UNIts
RFR	UNLOCK
RS232	UTILITY
RTS	VALue
RUNSAfter	VBArs

Appendix B: Reserved Words

VDELta
VERBoSe
VERTical
VERTMENU
VERTPOS
VERTSCALE
VIDeo
VOLts
WAVEform
WAVFrm
WFid
WFMOFF
WFMPre
WIDth
XINcr
XMUIt
XOff
XON
XUNit
XZEro
YMUIt
YOff
YUNit
YZEro
ZMUIt
ZOff
ZOOm
ZUNit
ZZEro

Appendix C: Factory Setup

The following listing is the instrument response to the concatenated command FACTORY;SET?. This response describes the factory default setup in detail.

```
:HEADER 1;
:VERBOSE 1;
:ALIAS:STATE 0;
:DATA:ENCDG RIBINARY;DESTINATION REFA;SOURCE REFA;
  START 1;STOP 2500;WIDTH 1;
:AUTORANGE OFF;
:LOCK NONE;
:ACQUIRE:MODE SAMPLE;NUMAVG 16;NUMENV 8;
  STATE 1;STOPAFTER RUNSTOP;
:DISPLAY:MODE SCOPE;FORMAT YT;STYLE VECTORS;
  PERSISTENCE 5.0E-1;GRATICULE FULL;TRIGT 1;
  INTENSITY 50;CONTRAST 50;
:SELECT:CH1 1;CH2 0;MATH 0;REFA 0;REFB 0;CONTROL CH1;
:CURSOR:FUNCTION OFF;VBARS:UNITS SECONDS;
  POSITION1 -1.0E-3;POSITION2 1.0E-3;SELECT CURSOR1;
:CURSOR:HBARS:POSITION1 -1.6E0;POSITION2 1.6E0;
  SELECT CURSOR1;
:CURSOR:PAIRED:POSITION1 -1.0E-3;POSITION2 1.0E-3;
  SELECT CURSOR1;
:HARDCOPY:FORMAT INTERLEAF;PORT RS232;LAYOUT PORTRAIT;
:HORIZONTAL:MODE MAIN;RECORDLENGTH 2500;
  POSITION 5.0E1;SCALE 5.0E-5;TRIGGER:POSITION 50;
:HORIZONTAL:MAIN:SCALE 5.0E-5;
:HORIZONTAL:DELAY:MODE RUNSAFTER;
  SCALE 5.0E-6;TIME:RUNSAFTER 2.0E-7;
```

```
:HORIZONTAL:REFA LOCK;REFB LOCK;
:MEASUREMENT:MEAS1:TYPE PERIOD;SOURCE1 CH1;STATE 0;
:MEASUREMENT:MEAS2:TYPE PERIOD;SOURCE1 CH1;STATE 0;
:MEASUREMENT:MEAS3:TYPE PERIOD;SOURCE1 CH1;STATE 0;
:MEASUREMENT:MEAS4:TYPE PERIOD;SOURCE1 CH1;STATE 0;
:MEASUREMENT:IMMED:TYPE PERIOD;SOURCE1 CH1;
:MEASUREMENT:GATING 0;METHOD HISTOGRAM;
  REFLEVEL:METHOD PERCENT;ABSOLUTE:HIGH 0.0E0;
  LOW 0.0E0;MID 0.0E0;
:MEASUREMENT:REFLEVEL:PERCENT:HIGH 9.0E1;LOW 1.0E1;
  MID 5.0E1;
:REFA:WAVEFORM 1;
:REFB:WAVEFORM 2;
:TRIGGER:MAIN:MODE AUTO;TYPE EDGE;
  LEVEL 0.0E0;HOLDOFF:VALUE 4.95E-7;
:TRIGGER:MAIN:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;
:TRIGGER:MAIN:PULSE:WHEN LESS;EQUAL 1.0E1;
  NOTEQUAL 1.0E1;HOLDOFF:VALUE 4.95E-7;
:TRIGGER:MAIN:PULSE:SOURCE CH1;POLARITY NEGATIVE;
  WIDTH 9.9E-8;
:TRIGGER:MAIN:VIDEO:SOURCE CH1;HOLDOFF:VALUE 4.95E-7;
:TRIGGER:MAIN:VIDEO:FIELD FIELD1;SCAN RATE1;
:CH1:SCALE 1.0E0;POSITION 0.0E0;OFFSET 0.0E0;
  COUPLING DC;BANDWIDTH FULL;PROBE:UNITS VOLTS;
  VOLTSCALE X10;CURRENTSCALE MVA1;
:CH1:INVERT 0;
:CH2:SCALE 1.0E0;POSITION 0.0E0;OFFSET 0.0E0;
  COUPLING DC;BANDWIDTH FULL;PROBE:UNITS VOLTS;
  VOLTSCALE X10;CURRENTSCALE MVA1;
:CH2:INVERT 0;
```

```
:MATH:DEFINE "Ch1 + Ch2";
:ZOOM:VERTICAL:SCALE 1.0E0;POSITION 0.0E0;
:POWER:STANDBY 10;BACKLIGHT 3;
:DMM:Cursors:FUNCTION OFF;VBARS:POSITION1 1.67E2;
    POSITION2 7.1E1;SELECT CURSOR1;
:DMM:Cursors:HBARS:POSITION1 -2.5E1;POSITION2 2.5E1;
    SELECT CURSOR1;
:DMM:Cursors:PAIRED:POSITION1 1.67E2;POSITION2 7.1E1;
    SELECT CURSOR1;
:DMM:MEASUREMENT:MEAS1:TYPE MAXIMUM;STATE 0;
:DMM:MEASUREMENT:MEAS2:TYPE MAXIMUM;STATE 0;
:DMM:MEASUREMENT:MEAS3:TYPE MAXIMUM;STATE 0;
:DMM:DISPLAY:DELTA OFF;GRATICULE FULL;STYLE THICK;
:DMM:FREQUENCY:REJECT 60;
:DMM:FUNCTION DCV;PROBE:UNITS VOLTS;CURRENTSCALE 1.0E-3;
:DMM:HOLD OFF;BEEP:CONTINUITY ON;NEWMAX OFF;
:DMM:LOGGER:MODE PEAKDETECT;SECDIV 3.0E1;
:DMM:RANGE:ACV 4.0E1;DCV 4.0E1;OHMS 4.0E4;AUTO OFF;
:DMM:RECALL:DATA LOGGER
```




Glossary and Index

Glossary

+/- Rocker

The general-purpose rocker button on the front panel that you can use to set parameters. The specific parameter assigned to the +/- rocker depends on other selections.

ASCII

Acronym for the American Standard Code for Information Interchange. Controllers transmit commands to the digitizing oscilloscope using ASCII character encoding.

Address

A 7-bit code that identifies an instrument on the communication bus. The digitizing oscilloscope must have a unique address for the controller to recognize and transmit commands to it.

Backus-Naur Form (BNF)

A standard notation system for command syntax diagrams. The syntax diagrams in this manual use BNF notation.

Controller

A computer or other device that sends commands to and accepts responses from the digitizing oscilloscope.

EOI

A mnemonic referring to the control line “End or Identify” on the GPIB interface bus. One of the two possible end-of-message terminators.

EOM

A generic acronym referring to the end-of-message terminator. The end-of-message terminator is either an EOI or the ASCII code for line feed (LF).

IEEE

Acronym for the Institute for Electrical and Electronic Engineers.

QuickBASIC

A computer language (distributed by Microsoft) that is based on the Beginner's All-Purpose Symbolic Instruction Code.

QuickC

A computer language (distributed by Microsoft) that is based on C.

RS-232

A serial, full-duplex, asynchronous communication port that follows ANSI/EIA/TIA-562-1989[1], ANSI/EIA/TIA-574-1990[2], and CCITT V.24-1989[3] standards.

TEKSecure

A Tektronix custom command that initializes both waveform and setup memories. This overwrites any previously stored data.

Index

A

- abbreviating, command, 2–4
- ACQUIRE?, 2–39
- ACQUIRE:MODE, 2–40
- ACQUIRE:NUMACQ?, 2–41
- ACQUIRE:NUMAVG, 2–42
- ACQUIRE:NUMENV, 2–43
- ACQUIRE:STATE, 2–44
- ACQUIRE:STOPAFTER, 2–45
- acquisition command group, 2–11, 2–12
- acquisition commands
 - ACQUIRE?, 2–39
 - ACQUIRE:MODE, 2–40
 - ACQUIRE:NUMACQ?, 2–41
 - ACQUIRE:NUMAVG, 2–42
 - ACQUIRE:NUMENV, 2–43
 - ACQUIRE:STATE, 2–44
 - ACQUIRE:STOPAFTER, 2–45
- ALIAS, 2–46
- alias commands
 - ALIAS, 2–46
 - ALIAS:CATALOG?, 2–47
 - ALIAS:DEFINE, 2–47
 - ALIAS:DELETE, 2–48
 - ALIAS:DELETE:ALL, 2–49
 - ALIAS:DELETE:NAME, 2–50
 - ALIAS:STATE, 2–50
- ALIAS:CATALOG?, 2–47
- ALIAS:DEFINE, 2–47
- ALIAS:DELETE, 2–48
- ALIAS:DELETE:ALL, 2–49
- ALIAS:DELETE:NAME, 2–50
- ALIAS:STATE, 2–50
- ALLEV?, 2–51
- argument, command, 2–2
- ASCII, 2–1

- AUTORANGE, 2–52
- AUTOSET, 2–53

B

- block, command argument, 2–10
- BNF (Backus-Naur form), 2–1
- break, 2–4
- BUSY?, 2–54

C

- *CAL?, 2–55, 2–57
- CALIBRATE:CONTINUE, 2–55
- CALIBRATE:FACTORY, 2–56
- CALIBRATE:STATUS, 2–57
- calibration and diagnostic
 - command group, 2–12
- calibration and diagnostic commands
 - *CAL?, 2–55, 2–57
 - CALIBRATE:CONTINUE, 2–55
 - CALIBRATE:FACTORY, 2–56
 - CALIBRATE:STATUS, 2–57
 - DIAG:RESULT:FLAG?, 2–92
 - DIAG:RESULT:LOG?, 2–93
 - DIAG:SELECT:ALL, 2–93
 - DIAG:STATE EXECUTE, 2–94
- CH<x>?, 2–58
- CH<x>:BANDWIDTH, 2–59
- CH<x>:COUPLING, 2–60
- CH<x>:POSITION, 2–61
- CH<x>:PROBE?, 2–62
- CH<x>:PROBE:
 - CURRENTSCALE, 2–63
- CH<x>:PROBE:UNITS, 2–63

- CH<x>:PROBE:
 - VOLTSCALE, 2-64
- CH<x>:SCALE, 2-65
- CH<x>:VOLTS, 2-66
- channel, command mnemonic, 2-7
- CH<x>, command mnemonic, 2-7
- clear status, 2-67
- CLEARMENU, 2-66
- *CLS, 2-67
- command
 - abbreviating, 2-4
 - argument, 2-2
 - block argument, 2-10
 - common, 2-19, 2-22
 - concatenating, 2-4
 - header, 2-2
 - message, 2-2
 - mnemonic, 2-2
 - query, 2-1
 - rules for forming, 2-1
 - separator, 2-2
 - set, 2-1
 - syntax, 2-1
 - BNF (Backus-Naur form), 2-1
- command argument
 - numeric, 2-8
 - quoted string, 2-9
- command group
 - acquisition, 2-11, 2-12
 - calibration and diagnostic, 2-12
 - cursor, 2-13
 - display, 2-14
 - DMM beeper, 2-35
 - DMM calibration, 2-32
 - DMM function range, 2-33
 - DMM logger, 2-36
 - DMM measurement, 2-33
 - DMM probe, 2-34
 - DMM save and recall, 2-37
 - hardcopy, 2-15
 - horizontal, 2-16
 - measurement, 2-17
 - miscellaneous, 2-19
 - miscellaneous DMM, 2-32
 - save and recall, 2-21
 - status and error, 2-22
 - trigger, 2-23
 - vertical, 2-25
 - waveform, 2-26
- command mnemonic
 - channel, 2-7
 - CH<x>, 2-7
 - cursor position, 2-7
 - math waveform, 2-7
 - MATH<x>, 2-7
 - measurement specifier, 2-7
 - MEAS<x>, 2-7
 - POSITION<x>, 2-7
 - reference waveform, 2-8
 - REF<x>, 2-8
 - waveform, 2-8
 - <wfm>, 2-8
- command status, 1-7
- command syntax, 2-1
 - BNF (Backus-Naur form), 2-1
- common command, 2-22
- common GPIB commands
 - *CAL?, 2-55, 2-57
 - *CLS, 2-67
- computers, connecting, 1-3
- concatenating, command, 2-4
- configuration, command query, 2-167
- Connecting to an RS-232
 - device, 1-1
- conventions, x
- cursor command group, 2-13

- cursor commands
 - CURSOR?, 2–68
 - CURSOR:FUNCTION, 2–68
 - CURSOR:HBARS?, 2–69
 - CURSOR:HBARS:
 - DELTA?, 2–70
 - CURSOR:HBARS:
 - POSITION<x>, 2–70
 - CURSOR:HBARS:
 - SELECT, 2–71
 - CURSOR:PAIRED?, 2–72
 - CURSOR:PAIRED:
 - HDELTA, 2–72
 - CURSOR:PAIRED:HPOS1, 2–73
 - CURSOR:PAIRED:HPOS2, 2–74
 - CURSOR:PAIRED:
 - POSITION1, 2–74
 - CURSOR:PAIRED:
 - POSITION2, 2–75
 - CURSOR:PAIRED:
 - SELECT, 2–76
 - CURSOR:PAIRED:
 - VDELTA, 2–76
 - CURSOR:VBARS, 2–77
 - CURSOR:VBARS:
 - DELTA?, 2–78
 - CURSOR:VBARS:
 - POSITION<x>, 2–78
 - CURSOR:VBARS:
 - SELECT, 2–79
 - CURSOR:VBARS:UNITS, 2–80
- cursor position, command
 - mnemonic, 2–7
- CURSOR?, 2–68
- CURSOR:FUNCTION, 2–68
- CURSOR:HBARS?, 2–69
- CURSOR:HBARS:DELTA?, 2–70
- CURSOR:HBARS:
 - POSITION<x>, 2–70
- CURSOR:HBARS:SELECT, 2–71
- CURSOR:PAIRED?, 2–72
- CURSOR:PAIRED:HDELTA, 2–72
- CURSOR:PAIRED:HPOS1, 2–73
- CURSOR:PAIRED:HPOS2, 2–74
- CURSOR:PAIRED:
 - POSITION1, 2–74
- CURSOR:PAIRED:
 - POSITION2, 2–75
- CURSOR:PAIRED:SELECT, 2–76
- CURSOR:PAIRED:VDELTA, 2–76
- CURSOR:VBARS, 2–77
- CURSOR:VBARS:DELTA?, 2–78
- CURSOR:VBARS:
 - POSITION<x>, 2–78
- CURSOR:VBARS:SELECT, 2–79
- CURSOR:VBARS:UNITS, 2–80
- CURVE, 2–81

D

- DATA, 2–82
- DATA:DESTINATION, 2–83
- DATA:ENCDG, 2–84
- DATA:SOURCE, 2–86
- DATA:START, 2–87
- DATA:STOP, 2–87
- DATA:TARGET, 2–88
- DATA:WIDTH, 2–89
- *DDT, 2–90
- DESE command, 2–91, 3–4
- DESER register, 2–91, 2–193, 3–4
- device clear, 2–4
- DIAG:RESULT:FLAG?, 2–92
- DIAG:RESULT:LOG?, 2–93
- DIAG:SELECT:ALL, 2–93
- DIAG:STATE EXECUTE, 2–94
- display command group, 2–14

- display commands
 - CLEARMENU, 2–66
 - DISPLAY?, 2–95
 - DISPLAY:CONTRAST, 2–96
 - DISPLAY:FORMAT, 2–96
 - DISPLAY:GRATICULE, 2–97
 - DISPLAY:MODE, 2–98
 - DISPLAY:PERSISTENCE, 2–99
 - DISPLAY:STYLE, 2–100
 - DISPLAY:TRIGT, 2–101
- DISPLAY?, 2–95
- DISPLAY:CONTRAST, 2–96
- DISPLAY:FORMAT, 2–96
- DISPLAY:GRATICULE, 2–97
- DISPLAY:MODE, 2–98
- DISPLAY:PERSISTENCE, 2–99
- DISPLAY:STYLE, 2–100
- DISPLAY:TRIGT, 2–101
- DMM beeper command
 - group, 2–35
- DMM beeper commands
 - DMM:BEEP:
 - CONTINUITY, 2–102
 - DMM:BEEP:NEWMAX, 2–102
- DMM calibrate commands
 - DMM:CALIBRATE:
 - CONTINUE, 2–103
 - DMM:CALIBRATE:
 - FACTORY, 2–103
 - DMM:CALIBRATE:
 - STATUS?, 2–104
- DMM calibration command
 - group, 2–32
- DMM function range command
 - group, 2–33
- DMM function range commands
 - DMM:RANGE:ACV, 2–135
 - DMM:RANGE:AUTO, 2–136
 - DMM:RANGE:DCV, 2–137
 - DMM:RANGE:OHMS, 2–137
- DMM logger command
 - group, 2–36
- DMM logger commands
 - DMM:CURSOR:
 - FUNCTION, 2–105
 - DMM:CURSOR:
 - HBARS?, 2–106
 - DMM:CURSOR:HBARS:
 - DELTA?, 2–106
 - DMM:CURSOR:HBARS:
 - POSITION<x>, 2–107
 - DMM:CURSOR:HBARS:
 - SELECT, 2–108
 - DMM:CURSOR:
 - PAIRED?, 2–108
 - DMM:CURSOR:PAIRED:
 - HDELTA?, 2–109
 - DMM:CURSOR:PAIRED:
 - HPOS1?, 2–110
 - DMM:CURSOR:PAIRED:
 - HPOS2?, 2–110
 - DMM:CURSOR:PAIRED:
 - POSITION1, 2–111
 - DMM:CURSOR:PAIRED:
 - POSITION2, 2–112
 - DMM:CURSOR:PAIRED:
 - SELECT, 2–112
 - DMM:CURSOR:PAIRED:
 - VDELTA?, 2–113
 - DMM:CURSOR:
 - VBARS, 2–114
 - DMM:CURSOR:VBARS:
 - DELTA?, 2–114
 - DMM:CURSOR:VBARS:
 - POSITION1, 2–115
 - DMM:CURSOR:VBARS:
 - POSITION2, 2–116
 - DMM:CURSOR:VBARS:
 - SELECT, 2–117
 - DMM:CURVE, 2–117

- DMM:LOGGER:MODE, 2–124
- DMM:LOGGER:SECDIV, 2–125
- DMM:WAVFRM?, 2–141
- DMM:WFMPRE?, 2–141
- DMM:WFMPRE:WFID?, 2–142
- DMM measurement command
 - group, 2–33
- DMM measurement commands
 - DMM:DELTA:VALUE, 2–118
 - DMM:MEASUREMENT?, 2–127
 - DMM:MEASUREMENT:AVERAGE?, 2–127
 - DMM:MEASUREMENT:CONTINUITY?, 2–128
 - DMM:MEASUREMENT:DELTA?, 2–128
 - DMM:MEASUREMENT:MAX?, 2–129
 - DMM:MEASUREMENT:MEAS<x>: STATE, 2–130
 - DMM:MEASUREMENT:MEAS<x>: TYPE, 2–130
 - DMM:MEASUREMENT:MEAS<x>: VALUE?, 2–132
 - DMM:MEASUREMENT:MIN?, 2–132
 - DMM:MEASUREMENT:PK2PK?, 2–133
 - DMM:VALUE?, 2–143
- DMM probe command group, 2–34
- DMM probe commands
 - DMM:PROBE?, 2–133
 - DMM:PROBE:
 - CURRENTSCALE, 2–134
 - DMM:PROBE:UNITS, 2–135
- DMM save and recall command
 - group, 2–37
 - DMM save and recall commands
 - DMM:RECALL:DATA, 2–138
 - DMM:SAVE:DATA, 2–139
- DMM:BEEP:
 - CONTINUITY, 2–102
- DMM:BEEP:NEWMAX, 2–102
- DMM:CALIBRATE:
 - CONTINUE, 2–103
- DMM:CALIBRATE:FACTORY, 2–103
- DMM:CALIBRATE:STATUS?, 2–104
- DMM:CORSOR:
 - FUNCTION, 2–105
- DMM:CORSOR:HBARS?, 2–106
- DMM:CORSOR:HBARS:
 - DELTA?, 2–106
- DMM:CORSOR:HBARS:POSITION<x>, 2–107
- DMM:CORSOR:HBARS:SELECT, 2–108
- DMM:CORSOR:PAIRED?, 2–108
- DMM:CORSOR:PAIRED:HDELTA?, 2–109
- DMM:CORSOR:PAIRED:HPOS1?, 2–110
- DMM:CORSOR:PAIRED:HPOS2?, 2–110
- DMM:CORSOR:PAIRED:POSITION1, 2–111
- DMM:CORSOR:PAIRED:POSITION2, 2–112
- DMM:CORSOR:PAIRED:SELECT, 2–112
- DMM:CORSOR:PAIRED:VDELTA?, 2–113
- DMM:CORSOR:VBARS, 2–114
- DMM:CORSOR:VBARS:
 - DELTA?, 2–114
- DMM:CORSOR:VBARS:POSITION1, 2–115

DMM:Cursors:VBARS:
 POSITION2, 2–116
DMM:Cursors:VBARS:
 SELECT, 2–117
DMM:CURVE, 2–117
DMM:DELTA:VALUE, 2–118
DMM:DISPLAY?, 2–119
DMM:DISPLAY:DELTA, 2–120
DMM:DISPLAY:
 GRATICULE, 2–120
DMM:DISPLAY:STYLE, 2–121
DMM:FREQUENCY:
 REJECT, 2–122
DMM:FUNCTION, 2–123
DMM:HOLD, 2–124
DMM:LOGGER:MODE, 2–124
DMM:LOGGER:SECDIV, 2–125
DMM:MEASUREMENT?, 2–127
DMM:MEASUREMENT:
 AVERAGE?, 2–127
DMM:MEASUREMENT:
 CONTINUITY?, 2–128
DMM:MEASURE-
 MENT:DELTA?, 2–128
DMM:MEASUREMENT:
 MAX?, 2–129
DMM:MEASUREMENT:
 MEAS<x>: STATE, 2–130
DMM:MEASUREMENT:
 MEAS<x>: TYPE, 2–130
DMM:MEASUREMENT:
 MEAS<x>: UNITS?, 2–131
DMM:MEASUREMENT:
 MEAS<x>:VALUE?, 2–132
DMM:MEASUREMENT:
 MIN?, 2–132
DMM:MEASUREMENT:
 PK2PK?, 2–133
DMM:PROBE?, 2–133
DMM:PROBE:
 CURRENTSCALE, 2–134

DMM:PROBE:UNITS, 2–135
DMM:RANGE:ACV, 2–135
DMM:RANGE:AUTO, 2–136
DMM:RANGE:DCV, 2–137
DMM:RANGE:OHMS, 2–137
DMM:RECALL:DATA, 2–138
DMM:RESET, 2–139
DMM:SAVE:DATA, 2–139
DMM:STATUS?, 2–140
DMM:VALUE?, 2–143
DMM:WAVFRM?, 2–141
DMM:WFMPRE?, 2–141
DMM:WFMPRE:WFID?, 2–142

E

EOM (end of message), 2–6
error message, programming
 interface, 3–13
*ESE, 2–144, 3–5
ESER register, 2–144, 2–193, 3–5
*ESR?, 2–145
*ESR? query, 3–2
event handling, 3–1
event query, 2–145, 2–146
event queue, 2–145, 2–146, 3–6
EVENT?, 2–145
EVMSG?, 2–146
EVQTY?, 2–147
example programs, 4–1

F

FACTORY, 2–147
factory setup, detailed
 description, C–1

G

GPIB, EOM (end of message), 2–6

H

HARDCOPY, 2–149

hardcopy command group, 2–15

hardcopy commands

 HARDCOPY, 2–149

 HARDCOPY:FORMAT, 2–150

 HARDCOPY:LAYOUT, 2–151

 HARDCOPY:PORT, 2–152

HARDCOPY:FORMAT, 2–150

HARDCOPY:LAYOUT, 2–151

HARDCOPY:PORT, 2–152

HDR, 2–153

HEADER, 2–153

header

 command, 2–2, 2–153

 included in query response,
 2–153, 2–237

horizontal command group, 2–16

horizontal commands

 HORIZONTAL?, 2–154

 HORIZONTAL:DELAY?, 2–155

 HORIZONTAL:DELAY:

 MODE, 2–156

 HORIZONTAL:DELAY:

 SCALE, 2–156

 HORIZONTAL:DELAY:

 SECDIV, 2–157

 HORIZONTAL:DELAY:

 TIME, 2–158

 HORIZONTAL:DELAY:

 TIME?, 2–158

 HORIZONTAL:DELAY:TIME:

 RUNSAFTER, 2–159

 HORIZONTAL:

 FITTOSCREEN, 2–159

HORIZONTAL:MAIN?, 2–160

HORIZONTAL:MAIN:

 SCALE, 2–161

HORIZONTAL:MAIN:

 SECDIV, 2–161

HORIZONTAL:MODE, 2–162

HORIZONTAL:

 POSITION, 2–163

HORIZONTAL:SCALE, 2–164

HORIZONTAL:SECDIV, 2–165

HORIZONTAL:

 TRIGGER?, 2–165

HORIZONTAL:TRIGGER:

 POSITION, 2–166

HORIZONTAL?, 2–154

HORIZONTAL:DELAY?, 2–155

HORIZONTAL:DELAY:

 MODE, 2–156

HORIZONTAL:DELAY:

 SCALE, 2–156

HORIZONTAL:DELAY:

 SECDIV, 2–157

HORIZONTAL:DELAY:

 TIME, 2–158

HORIZONTAL:DELAY:

 TIME?, 2–158

HORIZONTAL:DELAY:TIME:

 RUNSAFTER, 2–159

HORIZONTAL:

 FITTOSCREEN, 2–159

HORIZONTAL:MAIN?, 2–160

HORIZONTAL:MAIN:

 SCALE, 2–161

HORIZONTAL:MAIN:

 SECDIV, 2–161

HORIZONTAL:MODE, 2–162

HORIZONTAL:POSITION, 2–163

HORIZONTAL:SCALE, 2–164

HORIZONTAL:SECDIV, 2–165

HORIZONTAL:

TRIGGER?, 2-165

HORIZONTAL:TRIGGER:

POSITION, 2-166

I

ID?, 2-167

*IDN?, 2-167

IEEE Std. 488.2-1987, 1-1, 2-1,
2-19, 2-22

L

LOCK, 2-169

*LRN?, 2-169

M

manual conventions, x

manual trigger, simulation
with command, 2-218

math waveform, command
mnemonic, 2-7

MATH<x>?, 2-170

MATH<x>:DEFINE, 2-171

MATH<x>, command
mnemonic, 2-7

measurement command
group, 2-17

measurement commands

DMM:MEASUREMENT:

MEAS<x>: UNITS?, 2-131

MEASUREMENT?, 2-172

MEASUREMENT:

GATING, 2-172

MEASUREMENT:

IMMED?, 2-173

MEASUREMENT:IMMED:

SOURCE1, 2-174

MEASUREMENT:IMMED:

TYPE, 2-174

MEASUREMENT:IMMED:

UNITS?, 2-176

MEASUREMENT:IMMED:

VALUE?, 2-177

MEASUREMENT:

MEAS<x>?, 2-177

MEASUREMENT:MEAS<x>:

STATE, 2-179

MEASUREMENT:MEAS<x>:

TYPE, 2-179

MEASUREMENT:MEAS<x>:

UNITS?, 2-181

MEASUREMENT:MEAS<x>:

VALUE?, 2-182

MEASUREMENT:MEAS<x>:

SOURCE1, 2-178

MEASUREMENT:

METHOD, 2-183

MEASUREMENT:

REFLEVEL?, 2-183

MEASUREMENT:REFLEVEL:

ABSOLUTE:HIGH, 2-184

MEASUREMENT:REFLEVEL:

ABSOLUTE:MID, 2-185

MEASUREMENT:REFLEVEL:

METHOD, 2-186

MEASUREMENT:REFLEVEL:

PERCENT:HIGH, 2-187

MEASUREMENT:REFLEVEL:

PERCENT:LOW, 2-187

MEASUREMENT:REFLEVEL:

PERCENT:MID, 2-188

MEASUREMENT:REFLEVEL:

ABSOLUTE:LOW, 2-185

measurement specifier, command
mnemonic, 2-7

MEASUREMENT?, 2-172

MEASUREMENT:

GATING, 2-172

- MEASUREMENT:
 IMMED?, 2-173
- MEASUREMENT:IMMED:
 SOURCE1, 2-174
- MEASUREMENT:IMMED:
 TYPE, 2-174
- MEASUREMENT:IMMED:
 UNITS?, 2-176
- MEASUREMENT:IMMED:
 VALUE?, 2-177
- MEASUREMENT:
 MEAS<x>?, 2-177
- MEASUREMENT:
 MEAS<x>: STATE, 2-179
- MEASUREMENT:
 MEAS<x>: TYPE, 2-179
- MEASUREMENT:
 MEAS<x>: UNITS?, 2-181
- MEASUREMENT:
 MEAS<x>: VALUE?, 2-182
- MEASUREMENT:
 MEAS<x>:SOURCE1, 2-178
- MEASUREMENT:
 METHOD, 2-183
- MEASUREMENT:
 REFLEVEL?, 2-183
- MEASUREMENT:REFLEVEL:
 ABSOLUTE:HIGH, 2-184
- MEASUREMENT:REFLEVEL:
 ABSOLUTE:LOW, 2-185
- MEASUREMENT:REFLEVEL:
 ABSOLUTE:MID, 2-185
- MEASUREMENT:REFLEVEL:
 METHOD, 2-186
- MEASUREMENT:REFLEVEL:
 PERCENT:HIGH, 2-187
- MEASUREMENT:REFLEVEL:
 PERCENT:LOW, 2-187
- MEASUREMENT:REFLEVEL:
 PERCENT:MID, 2-188
- MEAS<x>, command
 mnemonic, 2-7
- message
 command, 2-2
 command terminator, 2-6
 handling, 3-1
 table of program messages, 3-13
- miscellaneous, LOCK, 2-169
- miscellaneous command
 group, 2-19
- miscellaneous commands
 AUTORANGE, 2-52
 AUTOSSET, 2-53
 *DDT, 2-90
 FACTORY, 2-147
 HDR, 2-153
 HEADER, 2-153
 *IDN?, 2-167
 *LRN?, 2-169
 NEWPASS, 2-189
 PASSWORD, 2-191
 POWER:BACKLIGHT, 2-192
 POWER:STANDBY, 2-192
 *PUD, 2-194
 REM, 2-198
 SET, 2-215
 TEKSECURE, 2-217
 *TRG, 2-218
 UNLOCK, 2-237
 VERBOSE, 2-237
- miscellaneous DMM
 command group, 2-32

miscellaneous DMM commands
DMM:DISPLAY?, 2-119
DMM:DISPLAY:DELTA, 2-120
DMM:DISPLAY:
GRATICULE, 2-120
DMM:DISPLAY:STYLE, 2-121
DMM:FREQUENCY:
REJECT, 2-122
DMM:FUNCTION, 2-123
DMM:HOLD, 2-124
DMM:RESET, 2-139
DMM:STATUS?, 2-140
mnemonic, command, 2-2

N

NEWPASS, 2-189
numeric, command argument, 2-8

O

*OPC, 2-189
operation complete
command, 2-189
operation complete wait, 2-238
oscilloscope save and recall
commands
REFA:WAVEFORM, 2-196
REFB:WAVEFORM, 2-197
output queue, 3-6

P

PASSWORD, 2-191
POSITION<x>, command
mnemonic, 2-7
POWER:BACKLIGHT, 2-192
POWER:STANDBY, 2-192
power-on status clear
command, 2-193

printers, connecting, 1-3
programming examples, 4-1
programming model, 1-10
*PSC, 2-193
*PSC command, 3-6
*PUD, 2-194

Q

query, header in query
response, 2-153, 2-237
query command, 2-1
queue
event, 3-6
output, 3-6
quoted string, command
argument, 2-9

R

*RCL, 2-195
recall setting command, 2-195
RECALL:SETUP, 2-196
REFA:WAVEFORM, 2-196
REFB:WAVEFORM, 2-197
reference waveform, command
mnemonic, 2-8
REF<x>, command
mnemonic, 2-8
register
DESER, 2-91, 2-193, 3-4
ESER, 2-144, 2-193, 3-5
SBR, 2-217, 3-3
SESR, 2-67, 2-145, 2-189, 3-2
SRER, 2-193, 2-216, 3-5
REM, 2-198
reserved words, B-1
reset
command, 2-209
factory, 2-147

- RS-232, 1–1
 - adapters, 1–3
 - BREAK, 1–6
 - command status, 1–7
 - configuring port, 1–4
 - Connecting to, 1–1
 - conventions, 1–6
 - default parameters, 1–5
 - errors, 1–6
 - transferring binary data, 1–6
 - troubleshooting, 1–7
 - RS232 commands
 - RS232?, 2–198
 - RS232:BAUD, 2–199
 - RS232:CONTROL:DCD, 2–199
 - RS232:CONTROL:RTS, 2–200
 - RS232:HARDFLAGGING, 2–201
 - RS232:MODE, 2–202
 - RS232:PACE, 2–202
 - RS232:PARITY, 2–203
 - RS232:PRESET, 2–204
 - RS232:SBITS, 2–205
 - RS232:SOFTFLAGGING, 2–205
 - RS232:STOPBITS, 2–206
 - RS232:TRANSMIT:
 - DELAY, 2–207
 - RS232:TRANSMIT:
 - TERMINATOR, 2–208
 - RS232?, 2–198
 - RS232:BAUD, 2–199
 - RS232:CONTROL:DCD, 2–199
 - RS232:CONTROL:RTS, 2–200
 - RS232:HARDFLAGGING, 2–201
 - RS232:MODE, 2–202
 - RS232:PACE, 2–202
 - RS232:PARITY, 2–203
 - RS232:PRESET, 2–204
 - RS232:SBITS, 2–205
 - RS232:SOFTFLAGGING, 2–205
 - RS232:STOPBITS, 2–206
 - RS232:TRANSMIT:
 - DELAY, 2–207
 - RS232:TRANSMIT:
 - TERMINATOR, 2–208
 - RS232:TRANSMIT:
 - DELAY, 2–207
 - RS232:TRANSMIT:
 - TERMINATOR, 2–208
 - *RST, 2–209
 - rules, command forming, 2–1
- ## S
- *SAV, 2–210
 - save and recall command
 - group, 2–21
 - save and recall commands
 - *RCL, 2–195
 - RECALL:SETUP, 2–196
 - *SAV, 2–210
 - SAVE:SETUP, 2–211
 - SAVE:WAVEFORM, 2–211
 - save setting command, 2–210
 - SAVE:SETUP, 2–211
 - SAVE:WAVEFORM, 2–211
 - SBR register, 2–217, 3–3
 - SELECT?, 2–213
 - SELECT:<wfm>, 2–213
 - SELECT:CONTROL?, 2–214
 - self test, 2–236
 - separator, command, 2–2
 - serial poll, 3–3
 - service request enable
 - command, 2–216
 - service request enable
 - register, 2–216
 - SESR register, 2–67, 2–145, 2–189, 3–2
 - set command, 2–1
 - SET?, 2–215

setting

- command query, 2–169
- query, 2–169
- recall command, 2–195
- save command, 2–210
- setups, factory setup
 - description, C–1
- *SRE command, 2–216, 3–5
- SRER register, 2–193, 2–216, 3–5
- status, 3–1
- status and error command
 - group, 2–22
- status and error commands
 - ALLEV?, 2–51
 - BUSY?, 2–54
 - *CLS, 2–67
 - DESE, 2–91, 3–4
 - *ESE, 2–144, 3–5
 - *ESR?, 2–145, 3–2
 - EVENT?, 2–145
 - EVMSG?, 2–146
 - EVQTY?, 2–147
 - ID?, 2–167
 - *OPC, 2–189
 - *PSC, 2–193, 3–6
 - *RST, 2–209
 - *SRE, 2–216, 3–5
 - *STB?, 2–217, 3–3
 - *TST?, 2–236
 - *WAI, 2–238
- *STB?, 2–217
- *STB? query, 3–3
- syntax
 - BNF (Backus-Naur form), 2–1
 - command, 2–1

T

- table, programming message, 3–13
- Tek Std. Codes and Formats 1989,
 - 2–19, 2–22
- TEKSECURE, 2–217
- terminator, command message, 2–6
- time base, manual trigger
 - simulation, 2–218
- *TRG, 2–218
- TRIGGER, 2–218
- trigger command group, 2–23
- trigger commands
 - TRIGGER, 2–218
 - TRIGGER:MAIN, 2–219
 - TRIGGER:MAIN:
 - EDGE?, 2–220
 - TRIGGER:MAIN:EDGE:
 - COUPLING, 2–221
 - TRIGGER:MAIN:EDGE:
 - SLOPE, 2–222
 - TRIGGER:MAIN:EDGE:
 - SOURCE, 2–222, 2–229
 - TRIGGER:MAIN:
 - HOLDOFF?, 2–223
 - TRIGGER:MAIN:HOLDOFF:
 - VALUE, 2–223, 2–227
 - TRIGGER:MAIN:
 - LEVEL, 2–224
 - TRIGGER:MAIN:MODE, 2–225
 - TRIGGER:MAIN:PULSE:
 - EQUAL, 2–225
 - TRIGGER:MAIN:PULSE:
 - HOLDOFF, 2–226
 - TRIGGER:MAIN:PULSE:
 - NOTEQUAL, 2–227
 - TRIGGER:MAIN:PULSE:
 - POLARITY, 2–228
 - TRIGGER:MAIN:PULSE:
 - WHEN, 2–229

- TRIGGER:MAIN:PULSE:
 - WIDTH, 2–230
 - TRIGGER:MAIN:TYPE, 2–231
 - TRIGGER:MAIN:VIDEO:
 - FIELD, 2–232
 - TRIGGER:MAIN:VIDEO:
 - HOLDOFF?, 2–232
 - TRIGGER:MAIN:VIDEO:
 - HOLDOFF:VALUE, 2–233
 - TRIGGER:MAIN:VIDEO:
 - SCAN, 2–234
 - TRIGGER:MAIN:VIDEO:
 - SOURCE, 2–234
 - TRIGGER:STATE?, 2–235
 - TRIGGER:MAIN, 2–219
 - TRIGGER:MAIN:EDGE?, 2–220
 - TRIGGER:MAIN:EDGE:
 - COUPLING, 2–221
 - TRIGGER:MAIN:EDGE:
 - SLOPE, 2–222
 - TRIGGER:MAIN:EDGE:
 - SOURCE, 2–222, 2–229
 - TRIGGER:MAIN:
 - HOLDOFF?, 2–223
 - TRIGGER:MAIN:HOLDOFF:
 - VALUE, 2–223, 2–227
 - TRIGGER:MAIN:LEVEL, 2–224
 - TRIGGER:MAIN:MODE, 2–225
 - TRIGGER:MAIN:PULSE:
 - EQUAL, 2–225
 - TRIGGER:MAIN:PULSE:
 - HOLDOFF, 2–226
 - TRIGGER:MAIN:PULSE:
 - NOTEQUAL, 2–227
 - TRIGGER:MAIN:PULSE:
 - POLARITY, 2–228
 - TRIGGER:MAIN:PULSE:
 - WHEN, 2–229
 - TRIGGER:MAIN:PULSE:
 - WIDTH, 2–230
 - TRIGGER:MAIN:TYPE, 2–231
 - TRIGGER:MAIN:VIDEO:
 - FIELD, 2–232
 - TRIGGER:MAIN:VIDEO:
 - HOLDOFF?, 2–232
 - TRIGGER:MAIN:VIDEO:
 - HOLDOFF:VALUE, 2–233
 - TRIGGER:MAIN:VIDEO:
 - SCAN, 2–234
 - TRIGGER:MAIN:VIDEO:
 - SOURCE, 2–234
 - TRIGGER:STATE?, 2–235
 - *TST? query, 2–236
 - tutorial, 1–10
- U**
- UNLOCK, 2–237
- V**
- VERBOSE, 2–237
 - vertical
 - MATH<x>?, 2–170
 - MATH<x>:DEFINE, 2–171
 - vertical bar cursors, 2–77
 - vertical command group, 2–25
 - vertical commands
 - CH<x>?, 2–58
 - CH<x>:BANDWIDTH, 2–59
 - CH<x>:COUPLING, 2–60
 - CH<x>:POSITION, 2–61
 - CH<x>:PROBE?, 2–62
 - CH<x>:PROBE:
 - CURRENTSCALE, 2–63
 - CH<x>:PROBE:UNITS, 2–63
 - CH<x>:PROBE:
 - VOLTSCALE, 2–64
 - CH<x>:SCALE, 2–65
 - CH<x>:VOLTS, 2–66
 - SELECT?, 2–213

SELECT:<wfm>, 2–213
 SELECT:CONTROL?, 2–214

W

*WAI, 2–238
 wait for operation complete, 2–238
 waveform, command
 mnemonic, 2–8
 waveform command group, 2–26
 waveform commands
 CURVE, 2–81
 DATA, 2–82
 DATA:DESTINATION, 2–83
 DATA:ENCDG, 2–84
 DATA:SOURCE, 2–86
 DATA:START, 2–87
 DATA:STOP, 2–87
 DATA:TARGET, 2–88
 DATA:WIDTH, 2–89
 WAVFRM?, 2–239
 WFMPRE?, 2–239
 WFMPRE:<wfm>?, 2–249
 WFMPRE:<wfm>:
 NR_PT, 2–250
 WFMPRE:<wfm>:
 PT_FMT, 2–250
 WFMPRE:<wfm>:
 PT_OFF, 2–251
 WFMPRE:<wfm>:WFID, 2–252
 WFMPRE:<wfm>:
 XINCR, 2–253
 WFMPRE:<wfm>:
 XUNIT, 2–253
 WFMPRE:<wfm>:
 YMULT, 2–254
 WFMPRE:<wfm>:YOFF, 2–255
 WFMPRE:<wfm>:
 YUNIT, 2–255
 WFMPRE:<wfm>:
 YZERO, 2–256

WFMPRE:BIT_NR, 2–240
 WFMPRE:BN_FMT, 2–241
 WFMPRE:BYT_NR, 2–241
 WFMPRE:BYT_OR, 2–242
 WFMPRE:ENCDG, 2–243
 WFMPRE:NR_PT, 2–248
 WFMPRE:PT_FMT, 2–244
 WFMPRE:PT_OFF, 2–245
 WFMPRE:WFID, 2–248
 WFMPRE:XINCR, 2–245
 WFMPRE:XMULT, 2–248
 WFMPRE:XOFF, 2–248
 WFMPRE:XUNIT, 2–248
 WFMPRE:XZERO, 2–248
 WFMPRE:YMULT, 2–246
 WFMPRE:YOFF, 2–246
 WFMPRE:YUNIT, 2–248
 WFMPRE:YZERO, 2–247
 WFMPRE:ZMULT, 2–248
 WFMPRE:ZOFF, 2–248
 WFMPRE:ZUNIT, 2–248
 WFMPRE:ZZERO, 2–248
 WAVFRM?, 2–239
 <wfm>, command mnemonic, 2–8
 WFMPRE?, 2–239
 WFMPRE:<wfm>?, 2–249
 WFMPRE:<wfm>:NR_PT, 2–250
 WFMPRE:<wfm>:PT_FMT, 2–250
 WFMPRE:<wfm>:PT_OFF, 2–251
 WFMPRE:<wfm>:WFID, 2–252
 WFMPRE:<wfm>:XINCR, 2–253
 WFMPRE:<wfm>:XUNIT, 2–253
 WFMPRE:<wfm>:YMULT, 2–254
 WFMPRE:<wfm>:YOFF, 2–255
 WFMPRE:<wfm>:YUNIT, 2–255
 WFMPRE:<wfm>:YZERO, 2–256
 WFMPRE:BIT_NR, 2–240
 WFMPRE:BN_FMT, 2–241
 WFMPRE:BYT_NR, 2–241
 WFMPRE:BYT_OR, 2–242
 WFMPRE:ENCDG, 2–243

WFMPRE:NR_PT, 2-248
WFMPRE:PT_FMT, 2-244
WFMPRE:PT_OFF, 2-245
WFMPRE:WFID, 2-248
WFMPRE:XINCR, 2-245
WFMPRE:XMULT, 2-248
WFMPRE:XOFF, 2-248
WFMPRE:XUNIT, 2-248
WFMPRE:XZERO, 2-248
WFMPRE:YMULT, 2-246
WFMPRE:YOFF, 2-246
WFMPRE:YUNIT, 2-248
WFMPRE:YZERO, 2-247
WFMPRE:ZMULT, 2-248
WFMPRE:ZOFF, 2-248
WFMPRE:ZUNIT, 2-248
WFMPRE:ZZERO, 2-248

Z

zoom commands
 ZOOM:STATE, 2-257
 ZOOM:VERTICAL:
 POSITION, 2-257
 ZOOM:VERTICAL:
 SCALE, 2-258
ZOOM:STATE, 2-257
ZOOM:VERTICAL:
 POSITION, 2-257
ZOOM:VERTICAL:
 SCALE, 2-258

