

**DL 1540C/1540CL**

Digital Oscilloscope  
Communication Interface

**USER'S MANUAL**

---

---

## Foreword

Thank you for purchasing YOKOGAWA's DL1540C/DL1540CL Digital Oscilloscope. This Communication Interface User's Manual describes the functions and commands of the GP-IB and RS-232-C interfaces, focusing on those used with the Model DL1540C Digital Oscilloscope. This manual also contains information on those functions that are not applicable or which are applicable exclusively to the DL1540CL digital oscilloscopes. To ensure proper use of the GP-IB/RS-232-C interface, please read this manual thoroughly.

Keep the manual in a safe place for quick reference whenever a question arises. Two manuals are provided with this instrument in addition to this Communication Interface User's Manual.

Manual Name	Manual No.	Description
DL1540C/DL1540CL User's Manual	IM 701530-01E	Describes all functions except for the communications functions and operation procedures of the instrument.
DL1540C/DL1540CL Operation Guide	IM 701530-02E	Describes the basic operations of the instrument.

## Note

The contents of this manual are subject to change without prior notice as a result of improvements in the instrument's performance and functions. Every effort has been made in the preparation of this manual to ensure the accuracy of its contents. However, should you have any questions or find any errors, please contact your nearest YOKOGAWA representative as listed on the back cover of this manual. Copying or reproduction of all or any part of the contents of this manual without YOKOGAWA's permission is strictly prohibited.

## Trademarks

HP-GL is a registered trademark of Hewlett-Packard Company.  
IBM PC/AT is a registered trademark of International Business Machines Corporation.

## Revisions

1st Edition: December 1998

# How to Use this Manual

## Structure of this Manual

This User's Manual consists of six chapters, an appendix and an index as described below.

### Chapter 1 Overview of the GP-IB Interface

Describes the functions and specifications of GP-IB.

### Chapter 2 Overview of the RS-232-C Interface

Describes the functions and specifications of RS-232-C.

### Chapter 3 Before Programming

Describes formats used when sending a command.

### Chapter 4 Commands

Describes each command.

### Chapter 5 Status Report

Describes the status byte, various registers and queues.

### Chapter 6 Sample Program

Describes a program example written for a IBM PC/AT series personal computer written in Quick BASIC version 4.0/4.5.

### Appendix

Contains references including the ASCII character code table.

### Index

Provides an alphabetically ordered index.

## Conventions Used in this Manual

Symbols used for Notes and Keys

Type	Symbol	Description
Unit	k	1000 e.g.: 100 kHz
	K	1024 e.g.: 128 KB (memory capacity)
Note	<b>Note</b>	Provides information that is necessary for proper operation of the instrument.

### Symbols used in syntax descriptions

Symbols which are used in the syntax descriptions in Chapter 4 are shown below. These symbols are referred to as BNF notation (Backus-Naur Form). For detailed information, refer to pages 3-5 and 3-6.

Symbol	Description	Example	Example
< >	Defined value	CHANnel<x> <x>=1, 2	CHANNEL2
{ }	One of the options in {} is selected.	MODE {AND OR}	MODE AND
	Exclusive OR	MODE {AND OR}	MODE AND
[ ]	Abbreviated	:MEASure[:MODE] {<NRf>}	
...	may be repeated		

# Contents

Foreword.....	1
How to Use this Manual.....	2
<b>Chapter 1 Overview of the GP-IB Function</b>	
1.1 Name and Function of Each Part .....	1-1
1.2 Connecting the GP-IB Cable .....	1-2
1.3 GP-IB Interface Functions and Specifications .....	1-2
1.4 Setting up this Instrument .....	1-4
1.5 Responses to Interface Messages .....	1-5
<b>Chapter 2 Overview of the RS-232-C Function</b>	
2.1 Name and Function of Each Part .....	2-1
2.2 RS-232-C Interface Functions and Specifications .....	2-2
2.3 Connecting the RS-232-C Interface Cable .....	2-2
2.4 Handshaking .....	2-4
2.5 Matching the Data Format .....	2-6
2.6 Setting up this Instrument .....	2-7
<b>Chapter 3 Before Programming</b>	
3.1 Messages .....	3-1
3.2 Commands .....	3-3
3.3 Response .....	3-4
3.4 Data .....	3-5
3.5 Synchronization with the Controller .....	3-7
<b>Chapter 4 Commands</b>	
4.1 List of Commands .....	4-1
4.2 ACQUIRE Group .....	4-11
4.3 ASETUP Group .....	4-12
4.4 CALIBRATE Group .....	4-12
4.5 CHANNEL Group .....	4-13
4.6 CLEAR Group .....	4-15
4.7 COMMUNICATE Group .....	4-15
4.8 CURSOR Group .....	4-17
4.9 DISPLAY Group .....	4-21
4.10 FILE Group .....	4-24
4.11 FILTER Group .....	4-28
4.12 GONOGO Group .....	4-29
4.13 HCOPI Group .....	4-36
4.14 IMAGE Group .....	4-45
4.15 INITIALIZE Group .....	4-45
4.16 MATH Group .....	4-46
4.17 MEASURE Group .....	4-48
4.18 RECALL Group .....	4-55
4.19 SCALE Group .....	4-56
4.20 SNAP Group .....	4-58

4.21	STARt Group .....	4-58
4.22	STATus Group .....	4-58
4.23	STOP Group .....	4-59
4.24	STORe Group .....	4-60
4.25	SYSTem Group .....	4-60
4.26	TIMEbase Group .....	4-62
4.27	TRIGger Group .....	4-63
4.28	UNIT Group .....	4-70
4.29	WAVEform Group .....	4-71
4.30	ZOOM Group .....	4-73
4.31	Common Command Group .....	4-74

**Chapter 5 Status Report**

5.1	Overview of the Status Report .....	5-1
5.2	Status Byte .....	5-2
5.3	Standard Event Register .....	5-3
5.4	Extended Event Register .....	5-4
5.5	Output Queue and Error Queue .....	5-5

**Chapter 6 Sample Program**

6.1	Before Programming .....	6-1
6.2	Setting the T/div .....	6-2
6.3	Reading the Latest Waveform Data of CH1 (Ascii Data) .....	6-3
6.4	Reading the Latest Waveform Data of CH1 (Binary Data) .....	6-5

**Appendix**

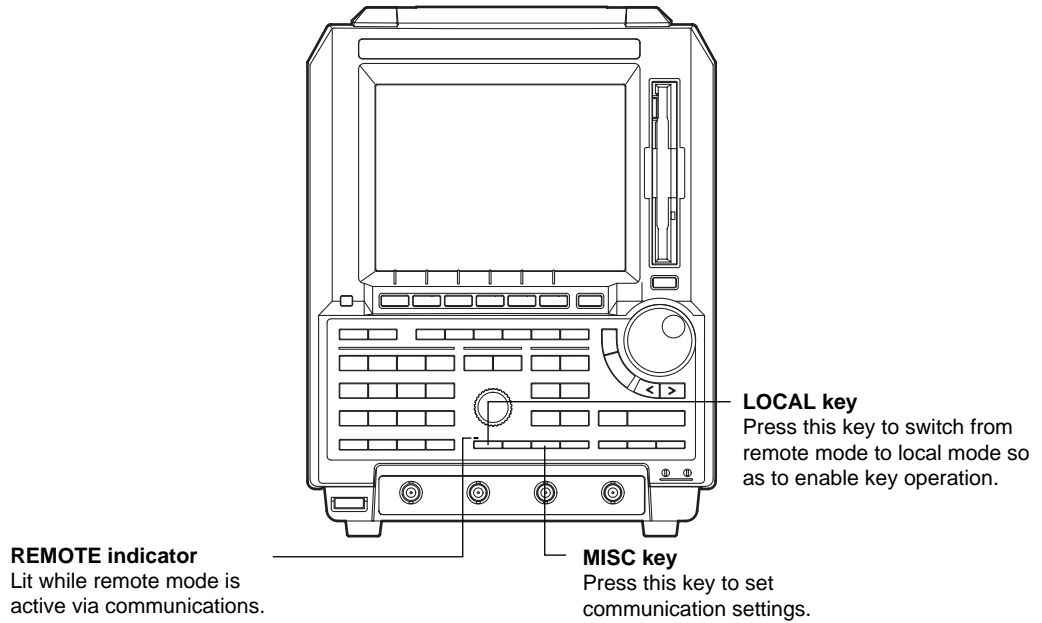
Appendix 1	ASCII Character Codes .....	App-1
Appendix 2	Error Messages .....	App-2
Appendix 3	Overview of IEEE 488.2-1992 .....	App-5

**Index**

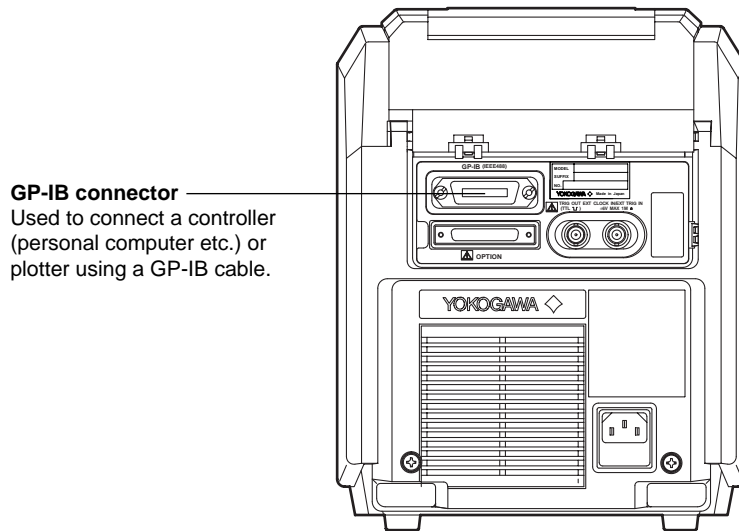
# 1 Overview of the GP-IB Function

## 1.1 Name and Function of Each Part

### Front Panel



**Rear Panel (Varies depending on the model. See the specifications of the respective models for details.)**



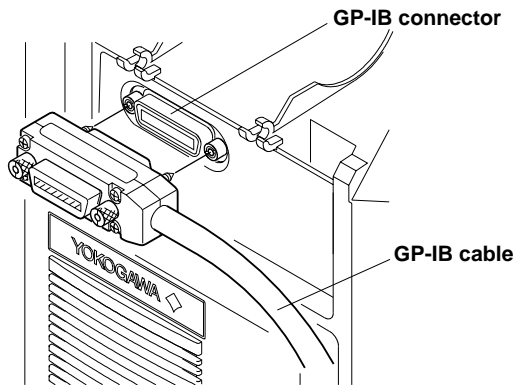
## 1.2 Connecting the GP-IB Cable

### GP-IB Cable

The GP-IB connector on the rear panel is a 24-pin connector that conforms to IEEE Standard 488-1978. Use a GP-IB cable that also conforms to IEEE Standard 488-1978.

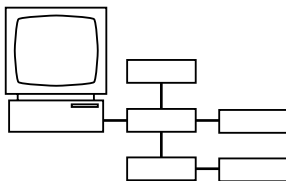
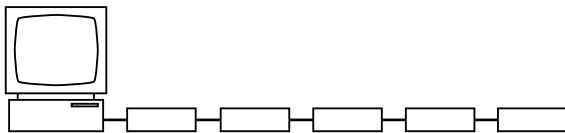
### Connection Method

Connect the GP-IB cable as shown.



### Points to Note

- Be sure to tighten the screws on the GP-IB cable connector firmly.
- The instrument can be connected to more than one item of equipment (such as a personal computer). However, it is not possible to connect more than 15 items of equipment (including the controller) to a single bus.
- If you connect the instrument to more than one other item of equipment, make sure that a different address is used for each item.
- Each connecting cable must be 2 m or less in length.
- The total length of all the cables must not exceed 20 m.
- While communications are in progress, more than two-thirds of the connected items of equipment must be turned ON.
- When connecting more than one item of equipment, connect the items so that the connection route forms a star or linear configuration. Loop or parallel wiring is not allowed.



## 1.3 GP-IB Interface Functions and Specifications

### GP-IB Interface Functions

#### Listener function

Allows you to make the same settings which you can make using the panel keys on the instrument (except for the power ON/OFF and GP-IB communications settings).

Receives commands from a controller requesting output of set-up and waveform data.

Also receives status report commands.

#### Talker function

Outputs set-up and waveform data.

#### Talk-only function

Outputs the screen data in various formats without using a controller. For more detailed information, refer to Sections 10.3 and 10.4 of the DL1540C/1540CL Digital Oscilloscope User's Manual. Talk-only mode is entered automatically when output is sent to a plotter. Output of waveform data is also possible to an AG Series Arbitrary Waveform Generator. For more details, refer to page 1-4.

#### Note

The listen-only and controller functions are not available on this instrument.

#### Switching from Local to Remote Mode

Remote mode is activated when a REN (Remote Enable) message is received from a controller while local mode is active.

- The REMOTE indicator lights up.
- All front panel keys except the LOCAL key are now inoperative.
- Settings that were entered in local mode are retained.

#### Switching from Remote to Local Mode

Pressing the Local key in remote mode puts the instrument in local mode. However, this is not possible if Local Lockout has been set by the controller (page 1-3).

- The REMOTE indicator goes out.
- All front panel keys are operative.
- Settings that were entered in remote mode are retained.

**GP-IB Interface Specifications**

- Electrical and mechanical specifications : Conforms to IEEE Standard 488-1978.
- Mechanical specifications : Refer to the table below.
- Protocol : Conforms to IEEE St'd 488.2-1992
- Code : ISO (ASCII) code
- Mode : Addressable mode/Talk-only mode (switched automatically)
- Address setting : Addresses 0 to 30 can be selected from the GP-IB setting screen, which is displayed when you press "Comm..." setting after having pressed the MISC key.
- Remote mode clear : Remote mode can be cleared by pressing the LOCAL key. However, this is not possible if Local Lockout has been set by the controller.

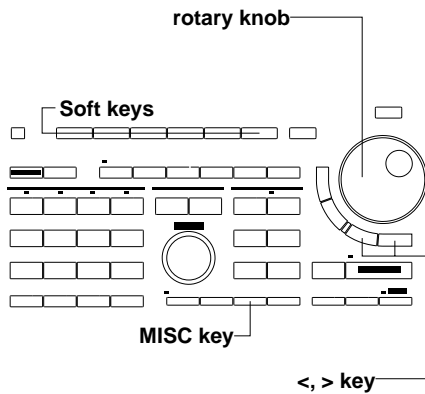
**Mechanical Specifications**

Function	Subset Name	Description
Source handshaking	SH1	Full source handshaking capability
Acceptor handshaking	AH1	Full acceptor handshaking capability
Talker	T5	Basic talker capability, serial polling, untalk on MLA(My Listen Address), talk-only capability
Listener	L4	Basic listener capability, unlisten on MTA (My Talk Address), no listen-only capability
Service request	SR1	Full service request capability
Remote local	RL1	Full remote/local capability
Parallel poll	PP0	No parallel polling capability
Device clear	DC1	Full device clear capability
Device trigger	DT0	No device trigger capability
Controller	C0	No controller function
Electrical characteristic	E1	Open collector



## 1.4 Setting up this Instrument

### Keys and Procedure



1. Press the **MISC key** to display the MISC top menu.
2. Press the “**Comm...**” soft key to display the interface selection menu.

Time Base...	Comm...	Date Time...	Infor- mation...	Config...	To NextMenu
--------------	---------	--------------	------------------	-----------	-------------

3. Press the “**GP-IB**” soft key.

Interface	
GP-IB	RS-232C

To TopMenu	GP-IB Address 1
------------	-----------------

#### Setting the Address

4. Turn the rotary knob to set the desired address.

#### Selecting Data Output (Talk-only Mode)

2. Press the “**To NextMenu**” soft key.
3. Press the “**DataOut...**” soft key to display the download menu.

DataOut...	Scale...	Self Test...	LCD...	To PrvMenu
------------	----------	--------------	--------	------------

4. Press the “**CH**” soft key to display the channel selection menu.

CH	Range		EXEC
1	T1	T2	

To TopMenu	T1 Cursor -5.00div
------------	--------------------

5. Set the downloading range by selecting either “**T1**” or “**T2**” soft key, and setting the vertical cursor position using the rotary knob. You can also use the < or > keys.

CH	Range		EXEC
2	T1	T2	

To TopMenu	T2 Cursor 2.70div
------------	-------------------

#### Executing the download function

6. Press the “**EXEC**” soft key. The name of the soft key will change to “**ABORT**”.

#### Aborting the download function

7. Press the “**ABORT**” soft key. The name of the soft key will change to “**EXEC**”.

### Explanation

#### Setting the Address

Each piece of equipment connected via a GP-IB interface has its own address by which it can be identified. Hence, this instrument’s address must be set when the instrument is to be connected to other items of equipment, such as a personal computer.

Set this instrument’s address within the range 0 to 30.

The default address is “1”.

#### Downloading setting: DataOut...

Downloading data from this instrument should only be done after having set downloading settings at the YOKOGAWA AG Series arbitrary waveform generator. For details regarding the settings and/or operation of the YOKOGAWA AG Series, refer to its corresponding instruction manual.

When executing downloading, the setting of this instrument will change to “talking mode” in the case of GP-IB. When downloading is finished or when the “ABORT” soft key is pressed, the previous setting will return.

The downloading range is set by the two cursors T1 and T2. The setting lies within – 5.00div at the left side to +5.00div at the right side measured from the center of the display.

## 1.5 Responses to Interface Messages

### What is an Interface Message?

An interface message (also called an interface command or bus command) is issued by the controller. Interface messages are classified as follows.

#### Uni-line messages

Messages are transferred through a single control line. The following three types of uni-line message are available.

- IFC (Interface Clear)
- REN (Remote Enable)
- IDY (Identify)

#### Multi-line messages

Eight data lines are used to transmit a message. Multi-line messages are classified as follows.

##### • Address commands

Valid when a piece of equipment is designated as a listener or a talker. The following five address commands are available.

##### • Commands valid for pieces of equipment designated as listeners

- GTL (Go To Local)
- SDC (Selected Device Clear)
- PPC (Parallel Poll Configure)
- GET (Group Execute Trigger)

##### • Command valid for pieces of equipment designated as talkers

- TCT (Take Control)

##### • Universal commands

Valid for any item of equipment, irrespective of whether the item is designated as a listener or a talker. The following five universal commands are available.

- LLO (Local Lockout)
- DCL (Device Clear)
- PPU (Parallel Poll Unconfigure)
- SPE (Serial Poll Enable)
- SPD (Serial Poll Disable)

In addition to the above commands, a listener address, talker address or secondary command can be sent in an interface message.

### Differences between SDC and DCL

The SDC command is an address command and requires that both the talker and listener be designated. DCL is a universal command and does not require that the talker and listener be designated. Hence, SDC is used for specific items of equipment, while DCL can be used for any equipment connected to the communications bus.

### Response to a uni-line message

- IFC (Interface Clear)  
Clears the talker and listener. Stops output if data is being output.
- REN (Remote Enable)  
Switches between remote and local modes.
- IDY (Identify) is not supported.

### Responses to a multi-line message (address command)

- GTL (Go To Local)  
Switches to local mode.
- SDC (Selected Device Clear)
  - Clears the program message (command) which is currently being output. Also empties the output queue (page 5-5).
  - \*OPC and \*OPC? will be disabled if they are currently being executed.
  - \*WAI and COMMunicate:WAIT will be stopped immediately.
- PPC (Parallel Poll Configure), GET (Group Execute Trigger) and TCT (Take Control) are not supported.

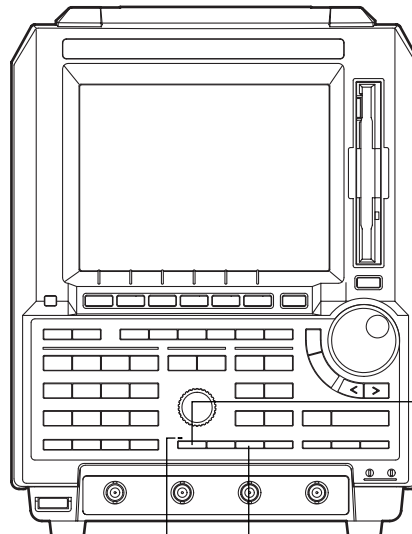
### Responses to a multi-line message (universal command)

- LLO (Local Lockout)  
Invalidates the LOCAL key on the front panel, disabling switching to local mode.
- DCL (Device Clear)  
Same as SDC.
- SPE (Serial Poll Enable)  
Sets the talker function to serial poll mode for all equipment connected to the communications bus. The controller polls equipment sequentially.
- SPD (Serial Poll Disable)  
Clears serial poll mode as the talker function for all equipment connected to the communications bus.
- PPU (Parallel Poll Unconfigure) is not supported.

# 2 Overview of the RS-232-C Function

## 2.1 Name and Function of Each Part

### Front Panel

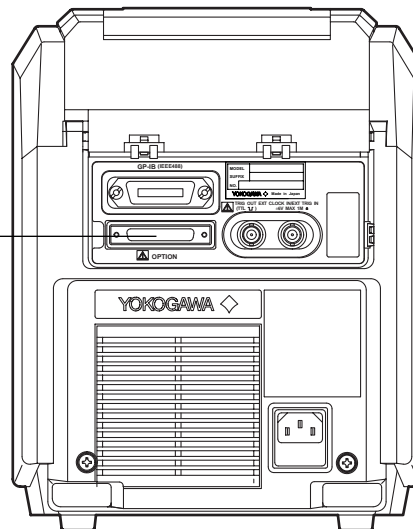


**REMOTE indicator**  
Lit while remote mode is active via communications.

**MISC key**  
Press this key to set communication settings.

**LOCAL key**  
Press this key to switch from remote mode to local mode so as to enable key operation.

**Rear Panel (Varies depending on the model. See the specifications of the respective models for details.)**



**Halfpitch interface connector**  
Used to connect the separately sold RS-232-C unit.

## 2.2 RS-232-C Interface Functions and Specifications

### Receiving Function

It is possible to make the same settings via the RS-232-C interface as can be made using the front panel keys.

Measured/computed data, panel set-up information and error codes can be received.

### Sending Function

Measured/computed data can be output.

Panel set-up information and the status byte can be output.

Error codes which have occurred can be output.

### RS-232C Interface Specifications

Electrical characteristics : Conforms to EIA RS-232-C.

Connection : Point-to-point

Communications : Full-duplex

Synchronization : Start-stop system

Baud rate : 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200

Start bit : 1 bit (fixed)

Data Length : 7 or 8 bits

Parity : Even, odd or no parity

Stop Bit : 1 or 2 bits

Connector : DELC-J9PAF-13L6 (JAE or equivalent)

Hardware handshaking : User can select whether CA or CB signals will always be True, or will be used for control.

Software Handshaking : User can select whether to control only transmission or both transmission and reception using X-on and X-off signals.  
X-on (ASCII 11H)  
X-off (ASCII 13H)

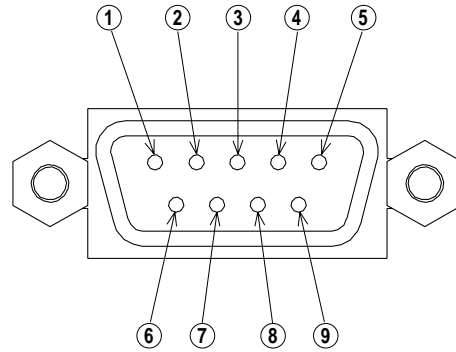
Receive buffer size : 256 bytes

## 2.3 Connecting the RS-232-C Interface Cable

When connecting this instrument to a computer, make sure that the handshaking method, data transmission rate and data format selected for the instrument match those selected for the computer.

For details, refer to the following pages. Also make sure that the correct interface cable is used.

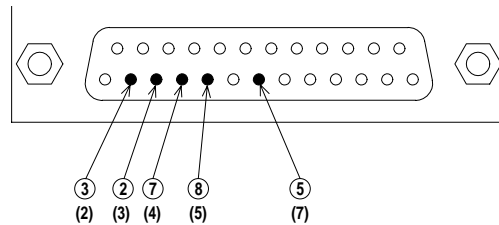
### Connector and Signal Names



- 2 RD (Received Data) : Data received from personal computer  
Signal direction ..... Input
- 3 SD (Send Data) : Data transmitted to a personal computer  
Signal direction ..... Output
- 5 SG (Signal Ground) : Ground for signals
- 7 RS (Request to Send) : Signal used for handshaking when receiving data from a personal computer  
Signal direction ..... Output
- 8 CS (Clear to Send) : Signal used for handshaking when transmitting data to a personal computer  
Signal direction ..... Input

Pin Nos. 1, 4, 6 and 9 are not used.

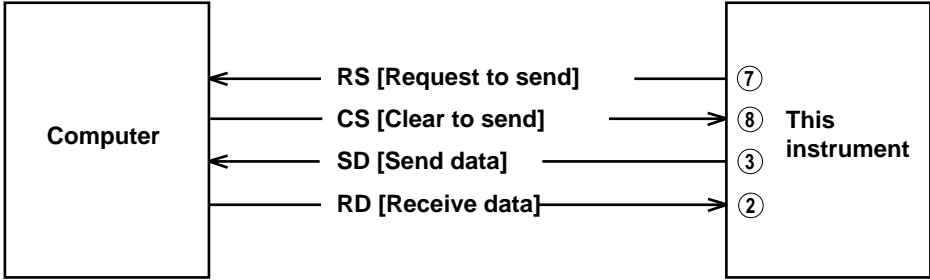
### 9-25 Pin Connector



The number between brackets refer to the pin Nos. of the 25-pin connector.

**Signal Direction**

The figure below shows the direction of the signals used by the RS-232-C interface.



**Table of RS-232-C Standard Signals and their JIS and CCITT Abbreviations**

Pin No. (9-pin connector)	Abbreviation			Description
	RS-232-C	CCITT	JIS	
⑤	AB (GND)	102	SG	Signal ground
③	BA (TXD)	103	SD	Transmitted data
②	BB (RXD)	104	RD	Received data
⑦	CA (RTS)	105	RS	Request to send
⑧	CB (CTS)	106	CS	Clear to send

## 2.4 Handshaking

To use an RS-232-C interface for transferring data between this instrument and a computer, it is necessary to use certain procedures by mutual agreement to ensure the proper transfer of data. These procedures are called "handshaking." Various handshaking systems are available depending on the computer to be used; the same handshaking system must be used for both the computer and this instrument.

This instrument allows you to choose any handshaking mode from the following four modes.

Handshake format Descriptions → ○

Handshake Method	Data Sending Control(control method when sending data to a computer)			Data Receiving Control(control method when receiving data from a computer)		
	Software Handshake	Hardware Handshake	No handshake	Software Handshake	Hardware Handshake	No handshake
	Sending stops when X-off is received, and sending is resumed when X-on is received.	Sending stops when CB(CTS) is False, and sending is resumed when CB is True.		X-off is sent when received data buffer becomes 3/4-full, and X-on is sent when the received data buffer is only 1/4-full.	CA (RTS) is set to False when received data buffer is only 3/4-full, and is set to True when received data buffer is only 1/4-full.	
OFF-OFF	NO-NO		○			○
XON-XON	XON-XON	○		○		
XON-RS	XON-RTS	○			○	
CS-RS	CTS-RTS		○		○	

The menu of this instrument

### 1 OFF-OFF

Transmission data control : There is no handshake status between the instrument and host computer. The X-OFF signal from the host computer is processed as data, and the CS signal is ignored.

Reception data control : There is no handshake status between the recorder and host computer. When the recorder reception buffer becomes full, the excess data is discarded. RS = True (fixed)

### Note

It is necessary to create a host computer program which prevents the buffers of both the instrument and the computer from becoming full.

### 2 XON-XON

Transmission data control : A software handshake status is established between the instrument and host computer. The instrument will stop a data transmission when an X-OFF signal is received from the host computer, and will resume transmission when the next X-ON signal is received. A CS signal from the host computer is ignored.

Reception data control : A software handshake status is established between the instrument and host computer. When the instruments reception buffer vacancy reaches 64bytes, the X-OFF signal will be sent to the host computer. When the reception buffer vacancy reaches 192 bytes, the X-ON signal will be sent. RS = True (fixed)

**3 XON-RS**

Transmission data control : A software handshake status is established between the instrument and host computer. The instrument will stop a data transmission when an X-OFF signal is received from the host computer, and will resume transmission when the next X-ON signal is received. A CS signal from the host computer is ignored.

Reception data control : A hardware handshake status is established between the instrument and host computer. When the instruments reception buffer vacancy reaches 64bytes, an "RS=False" status will be established. When the reception buffer vacancy reaches 192 bytes, an "RS=True" status will be established.

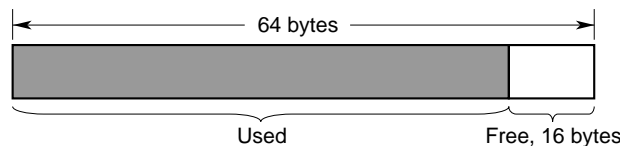
**4 CS-RS**

Transmission data control : A software handshake status is established between the instrument and host computer. The instrument will stop a data transmission if a "CS = False" status is established, and will resume the transmission shen a "CS = True" status is established. The X-OFF and X-ON signals from the host computer are processed as data.

Reception data control : A hardware handshake status is established between the instrument and host computer. When the intstruments reception buffer vacancy reaches 64bytes, an "RS=False" status will be established. When the reception buffer vacancy reaches 192 bytes, an "RS=True" status will be established.

**Precautions Regarding Data Receiving Control**

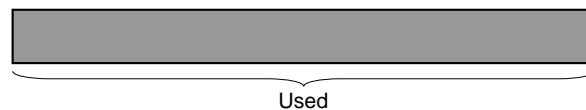
When handshaking is used to control the reception of data, data may still be sent from the computer even if the free space in the receive buffer drops below 64 bytes. In this case, after the receive buffer becomes full, the excess data will be lost, whether handshaking is in effect or not. Data storage to the buffer will begin again when there is free space in the buffer.



When handshaking is in use, reception of data will stop when the free space in the buffer drops to 16 bytes since data cannot be passed to the main program fast enough to keep up with the transmission.



After reception of data stops, data continues to be passed to the internal program. Reception of data starts again when the free space in the buffer increases to 48 bytes.

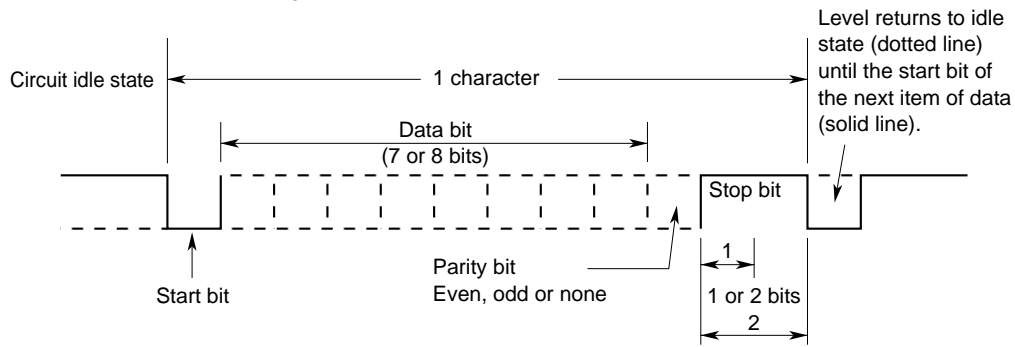


Whether handshaking is in use or not, if the buffer becomes full, any additional data received is no longer stored and is lost.

Data Receiving Control using Handshaking

## 2.5 Matching the Data Format

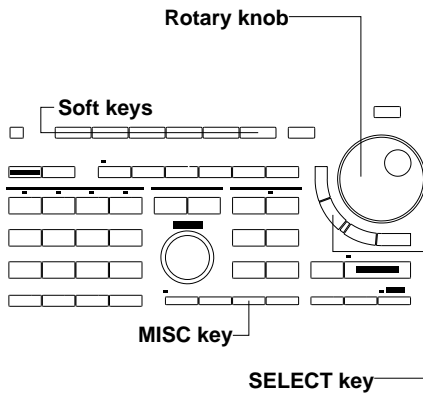
The RS-232-C interface of this instrument performs communications using start-stop synchronization. In start-stop synchronization, one character is transmitted at a time. Each character consists of a start bit, data bits, a parity bit and a stop bit. Refer to the figure below.





## 2.6 Setting up this Instrument

### Keys and Procedure

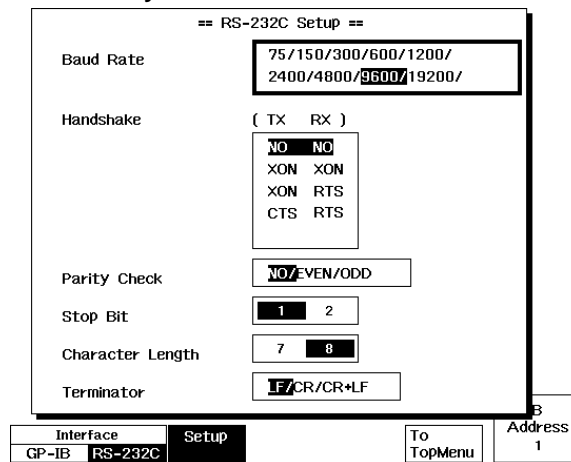


1. Press the **MISC key** to display the MISC menu.
2. Press the “**Comm...**” soft key to display the interface selection menu.
3. Press the “**RS-232C**” soft key, and then the “**Setup**” soft key to display the RS-232-C Setup menu.



### Setting the Baud rate and Data format

4. Turn the rotary knob to highlight the desired item and then press the **MISC key** to select the item.



### Explanation

#### Terminator

- In the menu above, select the terminator for sending data from this instrument. Select either “LF,” “CR,” or “CR+LF.”
- Use either “LF” or “CR+LF” as the terminator for receiving data.

# 3 Before Programming

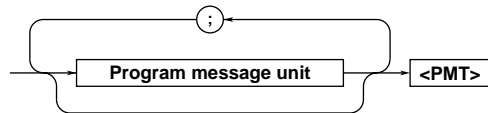
## 3.1 Messages

Blocks of message data are transferred between the controller and this instrument during communications. Messages sent from the controller to this instrument are called program messages, and messages sent back from this instrument to the controller are called response messages.

If a program message contains a query command, i.e. a command which requests a response, this instrument returns a response message. A single response message is always returned in reply to a program message.

### Program Messages

As explained above, the data (message) sent from the controller to this instrument is called a program message. The format of a program message is shown below.

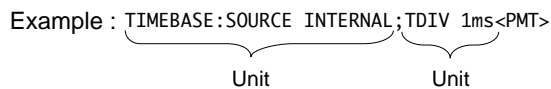


### <Program message unit>

A program message consists of zero or more program message units; each unit corresponds to one command. This instrument executes commands one by one according to the order in which they are received.

Program message units are delimited by a ";".

For a description of the format of the program message unit, refer to the explanation given further below.



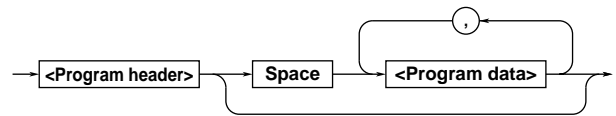
### <PMT>

PMT is a terminator used to terminate each program message. The following three types of terminator are available.

- NL (New Line) : Same as LF (Line Feed). ASCII code "0AH" is used.
- ^END : END message defined in IEEE488.1. (EOI signal) (The data byte sent with an END message will be the final item of the program message unit.)
- NL^END : NL with an END message attached (NL is not included in the program message unit.)

### Program message unit format

The format of a program message unit is shown below.

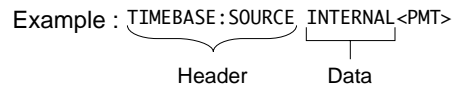


### <Program header>

A program header is used to indicate the command type. For details, refer to page 3-3.

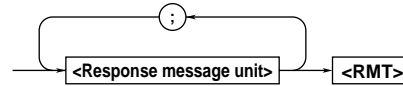
### <Program data>

If certain conditions are required for the execution of a command, program data must be added. Program data must be separated from the header by a space (ASCII code "20H"). If multiple items of program data are included, they must be separated by a "," (comma). For details, refer to page 3-5.



### Response Messages

The data returned by this instrument to the controller is called a response message. The format of a response message is shown below.

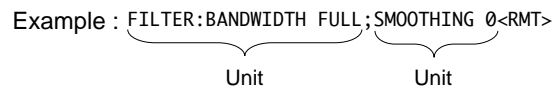


### <Response message units>

A response message consists of one or more response message units: each response message unit corresponds to one response.

Response message units are delimited by a ";".

For the response message format, refer to the next page.



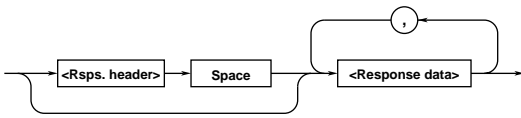
### <RMT>

RMT is the terminator used for every response message. Only one type of response message is available; NL^END.

### 3.1 Messages

#### Response message unit format

The format of a program message unit is shown below.



#### <Response header>

A response header sometimes precedes the response data. Response data must be separated from the header by a space. For details, refer to page 3-4.

#### <Response data>

Response data is used to define a response. If multiple items of response data are used, they must be separated by a “,” (comma). For details, refer to page 3-5.

Example :



If a program message contains more than one query, responses are made in the same order as the queries. Normally, each query returns only one response message unit, but there are some queries which return more than one response message unit. The first response message unit always responds to the first query, but it is not always true that the 'n'th unit always responds to the 'n'th query. Therefore, if you want to make sure that a response is made to each query, the program message must be divided up into individual messages.

#### Points to Note concerning Message Transmission

- It is always possible to send a program message if the previous message which was sent did not contain any queries.
- If the previous message contained a query, it is not possible to send another program message until a response message has been received. An error will occur if a program message is sent before a response message has been received in its entirety. A response message which has not been received will be discarded.
- If an attempt is made by the controller to receive a response message, even if there is no response message, an error will occur. An error will also occur if the controller makes an attempt to receive a response message before transmission of a program message has been completed.
- If a program message of more than one unit is sent and some of the units are incomplete, this instrument receives program message units which the instrument thinks complete and attempts to execute them. However, these attempts may not always be successful and a response may not always be returned, even if the program message contains queries.

#### Dead Lock

This instrument has a buffer memory in which both program and response messages of 1024 bytes or more can be stored. (The number of bytes available will vary depending on the operating state of the instrument.) If both buffer memories become full at the same time, this instrument becomes inoperative. This state is called dead lock. In this case, operation can be resumed by discarding the response message. No dead lock will occur, if the size of the program message including the PMT is kept below 1024 bytes. Furthermore, no dead lock will occur if the program message does not contain a query.

## 3.2 Commands

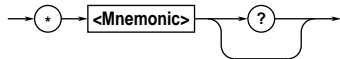
There are three types of command (program header) which can be sent from the controller to this instrument. They differ in the format of their program headers.

They are

- Common command header
- Compound header
- Simple header

### Common Command Header

Commands defined in IEEE 488.2-1987 are called common commands. The header format of a common command is shown below. An asterisk (\*) must always be attached to the beginning of a command.

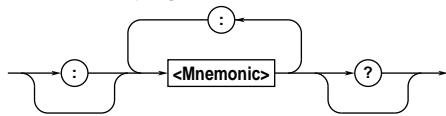


An example of a common command

\*CLS

### Compound Header

Commands designed to be used only with this instrument are classified and arranged in a hierarchy according to their function. The format of a compound header is illustrated below. A colon (:) must be used when specifying a lower-level header.

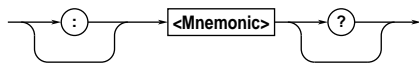


An example of a compound header

ACQuire:MODE NORMAL

### Simple Header

These commands (headers) are functionally independent of each other and are not arranged hierarchically. The format of a simple header is shown below.



An example of a simple header

START

### Note

A mnemonic is a character string made up of alphanumeric characters.

### When Concatenating Commands

#### Command Group

A command group is a group of commands which have the same compound header. A command group may contain sub-groups.

Example

#### Commands relating to computation

MATH?	MATH:FFT?
MATH:OPERator	MATH:FFT:MODE
MATH:PHASe?	MATH:FFT:TRACe
MATH:PHASe?:[:SHIFt]	MATH:FFT:WINDow
MATH:SENSitivity	MATH:FFT:DISPlay

#### When Concatenating Commands of the Same Group

This instrument stores the hierarchical level of the command which is currently being executed, and performs analysis on the assumption that the next command to be sent will also belong to the same level. Therefore, it is possible to omit the header if the commands belong to the same group.

Example ZOOM:MODE ZOOM;HZOOM 2<PMT>

#### When Concatenating Commands of Different Groups

A colon (:) must be included before the header of a command, if the command does not belong to the same group as the preceding command.

Example ZOOM:MODE ZOOM;:DISPLAY:XY ALL<PMT>

#### When Concatenating Simple Headers

When you type in a simple header after another command, you must include a colon (:) before the simple header.

Example ZOOM:MODE ZOOM;:START<PMT>

#### When Concatenating Common Commands

Common commands defined in IEEE 488.2-1992 are independent of hierarchical level. Thus, it is not necessary to add a colon (:) before a common command.

Example ZOOM:MODE ZOOM;\*CLS;HZOOM 2<PMT>

#### When Separating Commands with <PMT>

If a terminator is used to separate two commands, each command is a separate message. Therefore, the common header must be typed in for each command even when commands of the same command group are being concatenated.

Example ZOOM:MODE ZOOM<PMT>ZOOM:HZOOM 2<PMT>

### Upper-level Query

An upper-level query is a compound header to which a question mark is appended. Execution of an upper-level query allows all a group's settings to be output at once. Some query groups comprising more than three hierarchical levels can output all their lower level settings.

Example `FILTER?<PMT> :FILTER: BANDWIDTH  
FULL; SMOOTHING 0`

In reply to a query, a response can be returned as a program message to this instrument. Transmitting a response can restore the settings made when the query was executed. However, some upper-level queries will not return set-up data which is not currently in use. Note that not all a group's information will necessarily be sent out as a response.

### Header Interpretation Rules

This instrument interprets the header received according to the following rules.

- Mnemonics are not case sensitive.  
Example `"CURSor"` can also be written as `"cursor"` or `"Cursor"`.
- The lower-case part of a header can be omitted.  
Example `"CURSor"` can also be written as `"CURSO"` or `"CURS"`.
- If the header ends with a question mark, the command is a query. It is not possible to omit the question mark.  
Example `"CURSor?"` cannot be abbreviated to anything shorter than `"CURS?"`.
- If the "x" at the end of a mnemonic is omitted, it is assumed to be "1".  
Example If `"CHANne1<x>"` is written as `"CHAN"`, this represents `"CHANne11"`.
- Any part of a command enclosed by [ ] can be omitted.  
Example `CHANne1<x>[:INPut]:COUPling` can be written as `"CHAN:COUP"`.

However, a part enclosed by [ ] cannot be omitted if is located at the end of an upper-level query.

Example `"CHANne11?"` and `"CHANne11:INPut?"` belong to different upper-level query levels.

## 3.3 Response

On receiving a query from the controller, this instrument returns a response message to the controller. A response message is sent in one of the following two forms.

- Response consisting of a header and data  
If the query can be used as a program message without any change, a command header is attached to the query, which is then returned.  
Example `ACQuire:MODE?<PMT>→:ACQUIRE: MODE  
NORMAL<RMT>`
- Response consisting of data only  
If the query cannot be used as a program message unless changes are made to it (i.e. it is a query-only command), no header is attached and only the data is returned. Some query-only commands can be returned after a header is attached to them.  
Example `STATUS:ERROR?<PMT>→0,"NO ERROR"<RMT>`

### When returning a response without a header

It is possible to remove the header from a response consisting of a header and data. The `"COMMunicate:HEADer"` command is used to do this.

### Abbreviated form

Normally, the lower-case part is removed from a response header before the response is returned to the controller. Naturally, the full form of the header can also be used. For this, the `"COMMunicate:VERBose"` command is used. The part enclosed by [ ] is also omitted in the abbreviated form.

### 3.4 Data

#### Data

A data section comes after the header. A space must be included between the header and the data. The data contains conditions and values. Data is classified as below.

Data	Description
<Decimal>	Value expressed as a decimal number (Example: CH2's probe attenuation →CHANne12:PROBe 100)
<Voltage><Time><Frequency>	Physical value (Example: Time axis range →TIMebase:TDIV 1US)
<Register>	Register value expressed as either binary, octal, decimal or hexadecimal (Example: Extended event register value →STATus:EESE #HFE)
<Character data>	Specified character string (mnemonic). Can be selected from { } (Example: CH2 input coupling →CHANne12:INPut:COUPLing {ACIDC GND})
<Boolean>	Indicates ON/OFF. Set to ON, OFF or value (Example: CH2 display ON →CHANne12:MODE ON)
<Character string data>	Arbitrary character string (Example: File name to be saved →FILE:SAVE:SETup:NAME "SETUP_1")
<Block data>	Arbitrary 8-bit data (Example: Response to acquired waveform data →#6000010ABCDEFHIJ)

#### <Decimal>

<Decimal> indicates a value expressed as a decimal number, as shown in the table below. Decimal values are given in the NR form specified in ANSI X3. 42-1975.

Symbol	Description	Example
<NR1>	Integer	125 -1 +1000
<NR2>	Fixed point number	125.0 -.90 +001.
<NR3>	Floating point number	125.0E+0 -9E-1 +.1E4
<NRf>	Any of the forms <NR1> to <NR3> is allowed.	

Decimal values which are sent from the controller to this instrument can be sent in any of the forms to <NR3>. In this case, <NRf> appears.

For response messages which are returned from this instrument to the controller, the form (<NR1> to <NR3> to be used) is determined by the query. The same form is used, irrespective of whether the value is large or small.

In the case of <NR3>, the "+" after the "E" can be omitted, but the "-" cannot.

If a value outside the setting range is entered, the value will be normalized so that it is just inside the range.

If the value has more than the significant number of digits, the value will be rounded.

#### <Voltage>, <Time>, <Frequency>

<Voltage>, <Time> and <Frequency> indicate decimal values which have physical significance. <Multiplier> or <Unit> can be attached to <NRf>. They can be entered in any of the following forms.

Form	Example
<NRf><Multiplier><Unit>	5MV
<NRf><Unit>	5E-3V
<NRf><Multiplier>	5M
<NRf>	5E-3

#### <Multiplier>

Multipliers which can be used are shown below.

Symbol	Word	Description
EX	Exa	10 <sup>18</sup>
PE	Peta	10 <sup>15</sup>
T	Tera	10 <sup>12</sup>
G	Giga	10 <sup>9</sup>
MA	Mega	10 <sup>6</sup>
K	Kilo	10 <sup>3</sup>
M	Mili	10 <sup>-3</sup>
U	Micro	10 <sup>-6</sup>
N	Nano	10 <sup>-9</sup>
P	Pico	10 <sup>-12</sup>
F	Femto	10 <sup>-15</sup>
A	Atto	10 <sup>-18</sup>

#### <Unit>

Units which can be used are shown below.

Symbol	Word	Description
V	Volt	Voltage
S	Second	Time
HZ	Hertz	Frequency
MHZ	Megahertz	Frequency

<Multiplier> and <Unit> are not case sensitive.

"U" is used to indicate "μ".

"MA" is used for Mega (M) to distinguish it from Mili, except for in the case of Megahertz, which is expressed as "MHZ". Hence, it is not permissible to use "M" (Mili) for Hertz.

If both <Multiplier> and <Unit> are omitted, the default unit will be used.

Response messages are always expressed in <NR3> form. Neither <Multiplier> nor <Unit> is used, therefore the default unit is used.

### 3.4 Data

#### <Register>

<Register> indicates an integer, and can be expressed in hexadecimal, octal or binary as well as as a decimal number. <Register> is used when each bit of a value has a particular meaning. <Register> is expressed in one of the following forms.

Form	Example
<NRf>	1
#H<Hexadecimal value made up of the digits 0 to 9, and A to F>	#H0F
#Q<Octal value made up of the digits 0 to 7>	#q777
#B<Binary value made up of the digits 0 and 1>	#B001100

<Register> is not case sensitive.

Response messages are always expressed as <NR1>.

#### <Character Data>

<Character data> is a specified string of character data (a mnemonic). It is mainly used to indicate options, and is chosen from the character strings given in { }. For interpretation rules, refer to "Header Interpretation Rules" on page 3-4.

Form	Example
{AC DC GND}	AC

As with a header, the "COMMunicate:VERBose" command can be used to return a response message in its full form. Alternatively, the abbreviated form can be used. The "COMMunicate:HEADer" command does not affect <character data>.

#### <Boolean>

<Boolean> is data which indicates ON or OFF, and is expressed in one of the following forms.

Form	Example
{ON OFF <NRf>}	ON OFF 1 0

When <Boolean> is expressed in <NRf> form, OFF is selected if the rounded integer value is "0" and ON is selected if the rounded integer is "Not 0".

A response message is always "1" if the value is ON and "0" if it is OFF.

#### <Character String Data>

<Character string data> is not a specified character string like <Character data>. It is an arbitrary character string. A character string must be enclosed in single quotation marks (') or double quotation marks (").

Form	Example
<Character string data>	'ABC' "IEEE488.2-1987"

Response messages are always enclosed in double quotation marks.

If a character string contains a double quotation mark ("), the double quotation mark will be replaced by two concatenated double quotation marks ("). This rule also applies to a single quotation mark within a character string.

<Character string data> is an arbitrary character string, therefore this instrument assumes that the remaining program message units are part of the character string if no single (') or double quotation mark (") is encountered. As a result, no error will be detected if a quotation mark is omitted.

#### <Block data>

<Block data> is arbitrary 8-bit data. <Block data> is only used for response messages. Response messages are expressed in the following form.

Form	Example
#N<N-digit decimal value>	#6000010ABCDEFGHIJ
<Data byte string>	

#### #N

Indicates that the data is <Block data>. "N" is an ASCII character string number (digits) which indicates the number of data bytes that follow.

#### <N-digit decimal value>

Indicates the number of bytes of data. (Example:000010 = 10 bytes)

#### <Data byte string>

The actual data. (Example:ABCDEFGHIJ)

Data is comprised of 8-bit values (0 to 255). This means that the ASCII code "OAH", which stands for "NL", can also be a code used for data. Hence, care must be taken when programming the controller.

### 3.5 Synchronization with the Controller

#### Overlap Commands and Sequential Commands

There are two types of command groups: overlap commands and sequential commands.

In the case of overlap commands, an action by a subsequent command may take place before an action by a preceding command finishes.

For example, if you directly specify a V/div value and then send the programmed message shown below to inquire the result, the response is always the latest setpoint (5 V, in this example).

```
CHANne11:VDIV:VALue 5V;VALue?<PMT>
```

This is because the “CHANne11:VDIV:VALue” command forces subsequent commands to wait until it finishes its own processing. This type of command is called a sequential command. Alternatively, assume that you send the programmed message shown below when, for example, you have executed the automatic measurement of ACQ-waveform parameters and then inquire the result (AVERAGE for TRACE1 is set to ON in this example).

```
MEASure:ALLScan:EXECute;:MEASure:TRACe1:VALue?
AVERAge<PMT>
```

In that case, the “MEASure:TRACe1:VALue? AVERAge” command is executed before the automatic measurement ends, resulting in the response being the previous measured value. The action in which a command like “MEASure:ALLScan:EXECute” permits a subsequent command to run before it finishes its own processing is called an overlap operation. Commands that cause an overlap operation to occur are called overlap commands. Overlap operations can be avoided using one of the strategies described in the following paragraph.

#### Synchronization with an Overlap Command

##### • Using the \*WAI Command

The \*WAI command forces the commands that follow “\*WAI” to wait until an overlap command terminates.

```
Example:  COMMunicate:OPSE #H0080;:
          MEASure:ALLScan:EXECute;*WAI;:
          MEASure:TRACe1:VALue? AVERAge<PMT>
```

“COMMunicate:OPSE” is a command that designates which command or commands are included as the objects of the \*WAI command. In the above example, only the automatic measurement of ACQ-waveform parameters is targeted.

In the above message, the \*WAI command runs immediately before the “MEASure:TRACe1:VALue? AVERAge” command. The “MEASure:TRACe1: VALue? AVERAge” command, therefore, does not run until the automatic measurement of ACQ-waveform parameters ends.

##### • Using the COMMunicate:OVERlap Command

The “COMMunicate:OVERlap” command either permits or inhibits overlap operations.

```
Example:  COMMunicate:OVERlap #HFF7F;:
          MEASure:ALLScan:EXECute;:MEASure:TRACe1:
          VALue? AVERAge<PMT>
```

The “COMMunicate:OVERlap #HFF7F” command permits operations to overlap except for the automatic measurement of ACQ-waveform parameters. Since no overlap operation is permitted for the automatic measurement of ACQ-waveform parameters, the subsequent MEASure:ALLScan:EXECute command acts as a sequential command. The “MEASure:TRACe1:VALue? AVERAge” command, therefore, does not run until the automatic measurement ends.

##### • Using the \*OPC Command

The \*OPC command sets the OPC bit, which is bit 0 of the standard event register (see page 5-3), to “1” when an overlap operation ends.

```
Example:  COMMunicate:OPSE #H0080;*ESE
          1;*ESR?;*SRE 32;:MEASure:ALLScan:
          EXECute;*OPC<PMT>
          (Read the response to the *ESR? query.)
          (Wait for a service request.)
          MEASure:TRACe1:VALue? AVERAge<PMT>
```

“COMMunicate:OPSE” is a command that designates which command or commands are included as the objects of the \*OPC command. In this example, only the automatic measurement of waveform parameters is targeted.

\*ESE 1 and \*SRE 32 show that a service request is issued only when the OPC bit becomes “1”.

\*ESR? clears the standard event register. In the example, the “MEASure:TRACe1:VALue? AVERAge” command does not run until a service request is generated.



#### • Using the \*OPC? Query

The \*OPC? query is a command that generates a response when an overlap operation ends.

Example:      COMMunicate:OPSE #H0080; ;  
                 MEASure:ALLScan:EXECute;\*OPC?<PMT>  
                 (Read the response to the \*OPC? query.)  
                 MEASure:TRACe1:VALue? AVERAge<PMT>

“COMMunicate:OPSE” is a command that designates which command or commands are included as the objects of the \*OPC? query command. In this example, only the automatic measurement of waveform parameters is targeted.

Since the \*OPC? query does not generate a response until an overlap operation ends, the automatic measurement is complete by the time you read the response to the \*OPC? query.

#### **Note**

---

Most commands are sequential commands. Commands used in Chapter 4 are sequential commands unless otherwise specified.

---

#### **Synchronization with Non-Overlap Commands**

Synchronization is sometimes required for reasons other than communications-related reasons, such as the activation of a trigger, even if a sequential command is used.

For example, if a “next program” message is transmitted to make an inquiry about the waveform data which has been acquired using single mode as the trigger mode, the “WAVeform:SEND?” command is sometimes executed whether acquisition has been completed or not, causing a command execution error.

```
TRIGger:MODE SSINGle;:START;:WAVeform: SEND?<PMT>
```

In this case, synchronization with the time at which acquisition is completed must be accomplished, as shown next.

#### **Using STATus:CONDition? query**

A “STATus:CONDition?” query is used to make an inquiry about the contents of the condition register (page 5-4). It is possible to judge whether acquisition is in progress or not by reading bit 0 of the condition register. Bit 0 is “1” if acquisition is in progress, and “0” if acquisition is stopped.

Example      TRIGger:MODE SSINGle;:START<PMT>  
                 STATus:CONDition?<PMT>  
                 (Returns to the previous status if bit 0 is  
                 found to be “1” when the response is  
                 decoded.)  
                 WAVeform:SEND?<PMT>

A “WAVeform:SEND?” query will not be executed until bit 0 of the condition register has been set to “0”.

#### **Using the extended event register**

Changes in the condition register are reflected in the extended event register (page 5-4).

Example      STATus:FILTer1 FALL;:STATus:EES  
                 1;EESR?;\*SRE8;:TRIGger:MODE  
                 SSINGle;:START<PMT>  
                 (Service request is awaited.)  
                 WAVeform:SEND?<PMT>

“STATus:FILTer1 FALL” indicates that the transit filter is set so that bit 0 is set to “1” when bit 0 (FILTer 1) of the condition register is changed from “1” to “0”.

“STATus:EES 1” is a command used only to reflect the status of bit 0 of the extended event register in the status byte.

“STATus:EESR?” is used to clear the extended event register.

The “\*SRE” command is used to generate a service request caused solely by the extended event register. “WAVeform:SEND?” will not be executed until a service request is generated.

#### **Using the COMMunicate:WAIT command**

The “COMMunicate:WAIT” command halts communications until a specific event is generated.

Example      STATus:FILTer1 FALL;:STATus:EES  
                 1;EESR?;:TRIGger:MODE  
                 SSINGle;:START<PMT>  
                 (Response to STATus:EESR? is decoded.)  
                 COMMunicate:WAIT  
                 1;:WAVeform:SEND?<PMT>

For a description of “STATus:FILTer 1 FALL” and “STATus:EESR?”, refer to “Using the extended event register” on this page.

“COMMunicate:WAIT 1” means that communications is halted until bit 0 of the extended event register is set to “1”.

“WAVeform:SEND?” will not be executed until bit 0 of the extended event register is set to “1”.

# 4 Commands

## 4.1 List of Commands

Command	Description	Page
<b>ACQUIRE Group</b>		
:ACQUIRE?	Queries all the acquisition settings	4-11
:ACQUIRE:HISTORY	Sets the history memory function ON/OFF, or queries the current setting	4-11
:ACQUIRE[:MODE]	Sets the acquisition mode, or queries the current setting	4-11
:ACQUIRE:RECORDLENGTH?	Queries the record length (available on DL1540C)	4-11
:ACQUIRE:RECORDLENGTH	Sets the record length, or queries the current setting (available on DL1540CL)	4-11
:ACQUIRE:REPETITIVE	Sets/queries the record length	4-11
<b>ASETUP Group</b>		
:ASETUP:EXECUTE	Executes auto set-up	4-12
:ASETUP:UNDO	Cancels auto set-up settings	4-12
<b>CALIBRATE Group</b>		
:CALIBRATE?	Queries all the calibration settings	4-12
:CALIBRATE:EXECUTE	Executes calibration	4-12
:CALIBRATE:MODE	Turns calibration AUTO/OFF, or queries the current setting	4-12
<b>CHANNEL Group</b>		
:CHANNEL<x>?	Queries all the vertical axis settings for the specified channel	4-13
:CHANNEL<x>:INPUT?	Queries all the vertical axis input conditions for the specified channel	4-13
:CHANNEL<x>[:INPUT]:COUPLING	Sets input coupling for the specified channel, or queries the current setting	4-14
:CHANNEL<x>[:INPUT]:HOLD	Turns acquisition hold ON/OFF for the specified channel, or queries the current setting	4-14
:CHANNEL<x>[:INPUT]:INVERT	Turns inversion of the waveform ON/OFF for the specified channel, or queries the current setting	4-14
:CHANNEL<x>[:INPUT]:OFFSET	Sets the offset voltage for the specified channel, or queries the current setting	4-14
:CHANNEL<x>[:INPUT]:PROBE	Sets the probed attenuation for the specified channel, or queries the current setting	4-14
:CHANNEL<x>:MODE	Turns display ON/OFF for the specified channel, or queries the current setting	4-14
:CHANNEL<x>:POSITION	Sets the vertical position for the specified channel, or queries the current setting	4-14
:CHANNEL<x>:VDIV?	Queries all the voltage axis sensitivity settings for the specified channel	4-14
:CHANNEL<x>:VDIV:VALUE	Sets the voltage axis sensitivity for the specified channel, or queries the current setting	4-14
:CHANNEL<x>:VDIV:VARIABLE	Turns fine adjustment ON/OFF for the specified channel, or queries the current setting	4-14
<b>CLEAR Group</b>		
:CLEAR	Clears the trace	4-15
<b>COMMUNICATE Group</b>		
:COMMUNICATE?	Queries all the communication settings	4-15
:COMMUNICATE:HEADER	Determines whether a header is to be added or not	4-15
:COMMUNICATE:LOCKOUT	Sets local lockout ON or OFF	4-16
:COMMUNICATE:OPSE	Sets the overlap command to be used with *OPC, *OPC? and *WAI, or queries the current setting	4-16
:COMMUNICATE:OPSR?	Returns the value in the operation pending status register	4-16
:COMMUNICATE:OVERLAP	Selects the command to be used for overlap, or queries the current setting	4-16
:COMMUNICATE:REMOTE	Sets remote (ON) or local mode (OFF)	4-16
:COMMUNICATE:STATUS?	Queries the status of a specified circuit	4-16
:COMMUNICATE:VERBOSE	Determines whether a response to a query is to be returned in full or abbreviated form	4-16
:COMMUNICATE:WAIT	Waits until one of the specified extended event occurs	4-16
:COMMUNICATE:WAIT?	Generates a response when one of the specified extended events occurs	4-16
<b>CURSORS Group</b>		
:CURSOR?	Queries all the settings relating to the FFT cursors	4-18
:CURSOR:FFT?	Queries all the FFT cursor measurement settings	4-18
:CURSOR:FFT:DF?	Queries the $\Delta F$ value between the FFT cursors	4-18
:CURSOR:FFT:DP?	Queries the $\Delta P$ value between the FFT cursors	4-18
:CURSOR:FFT:F<x>?	Queries the frequency values of the FFT cursors	4-18

## 4.1 List of Commands

Command	Description	Page
:CURSor:FFT:P<x>?	Queries the P values of the FFT cursors	4-18
:CURSor:FFT:POSition<x>	Sets the position of the FFT cursor, or queries the current setting	4-18
:CURSor[:MODE]	Sets cursor measurement ON/OFF, or queries the current setting	4-18
:CURSor:TY?	Queries all the T-Y cursor measurement settings	4-18
:CURSor:TY:DDEGree?	Queries the phase difference value, $\Delta D$ , between D1 and D2 phase cursors	4-18
:CURSor:TY:DEGRee?	Queries the value of D1 or D2 cursor	4-18
:CURSor:TY:DT?	Queries the $\Delta T$ value between the T-Y cursors	4-18
:CURSor:TY:DV?	Queries the $\Delta V$ value between the T-Y cursors	4-19
:CURSor:TY:PERDt?	Queries the $1P/\Delta T$ value between the T-Y cursors	4-19
:CURSor:TY:POSition<x>	Sets the position of the T-Y cursor, or queries the current setting	4-19
:CURSor:TY:REFerence	Sets the position of the start point or end point (R1, R2) for the phase reference, or queries the current setting	4-19
:CURSor:TY:RVALue	Sets the phase reference value between the vertical cursors (phase cursors), or queries the current setting	4-19
:CURSor:TY:TRACe	Sets the waveform of measurement of the T-Y cursor, or queries the current setting	4-19
:CURSor:TY:TYPE	Sets the measurement type of the T-Y cursor, or queries the current setting	4-19
:CURSor:TY:T<x>?	Queries the time corresponding to the T-Y cursor	4-19
:CURSor:TY:V<x>?	Queries the voltage value corresponding to the T-Y cursor	4-19
:CURSor:XY?	Queries all the X-Y cursor measurement settings	4-19
:CURSor:XY:DX?	Queries the $\Delta X$ value between the X-Y cursors	4-19
:CURSor:XY:DY?	Queries the $\Delta Y$ value between the X-Y cursors	4-19
:CURSor:XY:POSition<x>	Sets the position of the X-Y cursor, or queries the current setting	4-19
:CURSor:XY:T?	Queries the time corresponding to the X-Y cursor	4-20
:CURSor:XY:TYPE	Sets the measurement type of the X-Y cursor, or queries the current setting	4-20
:CURSor:XY:X?	Queries the voltage value corresponding to the X waveform of the X-Y cursor	4-20
:CURSor:XY:Y?	Queries the voltage value corresponding to the Y waveform of the X-Y cursor	4-20
<b>DISPlay Group</b>		
:DISPlay?	Queries all the display settings	4-22
:DISPlay:ACCumulate?	Queries all the accumulation settings	4-22
:DISPlay:ACCumulate:COLor	Sets the color grade width of the waveform, or queries the current setting	4-22
:DISPlay:ACCumulate:MODE	Turns accumulation mode ON/OFF, or queries the current setting	4-22
:DISPlay:ACCumulate:PERSistance	Sets the accumulation time, or queries the current setting	4-22
:DISPlay:ALLexec	Displays all stored waveforms	4-22
:DISPlay:COUNt	Sets the number of the waveform to be displayed, or queries the current setting	4-22
:DISPlay:GRATicule	Sets the graticule, or queries the current setting	4-23
:DISPlay:INTENSity?	Queries the intensity of displayed parameters	4-23
:DISPlay:INTENSity:{CURSor SCALE SNAPshot TEXT WAVE}	Sets the intensity of each displayed parameter, or queries the current setting	4-23
:DISPlay:INTERpolate	Sets the interpolation method, or queries the current setting	4-23
:DISPlay:PMARker	Sets the % marker ON/OFF, or queries the current setting	4-23
:DISPlay:POSition	Sets the horizontal/vertical position of the waveform, or queries the current setting	4-23
:DISPlay:SCALE	Turns display of the scale ON/OFF, or queries the current setting	4-23
:DISPlay:TIME?	Queries the time of acquisition corresponding to the waveform with the display No.	4-23
:DISPlay:WINformation	Turns display of waveform information ON/OFF, or queries the current setting	4-23
:DISPlay:WSCreen	Turns the wide screen mode ON/OFF, or queries the current setting	4-23
:DISPlay:XY	Turns display of the X-Y waveform ON/OFF, or queries the current setting	4-23
<b>FILE Group</b>		
:FILE?	Queries all the settings relating to files	4-25
:FILE:ANAMing	Sets the auto naming mode for file names, or queries the current setting	4-25
:FILE:DELeTe:ACQuisition:EXECute	Executes the deletion of an acquisition data file	4-25
:FILE:DELeTe:ACQuisition:NAME	Sets the name of the acquisition data file to be deleted	4-25
:FILE:DELeTe:SETup:EXECute	Executes the deletion of a setting parameter data file	4-25
:FILE:DELeTe:SETup:NAME	Sets the name of the setting parameter file to be deleted	4-25
:FILE:DELeTe:TRACe:EXECute	Executes the deletion of a displayed waveform file	4-25
:FILE:DELeTe:TRACe:NAME	Sets the name of the displayed waveform file to be deleted	4-25
:FILE:DRIVE	Sets the target media to execute the FILE Group Commands, or queries the current setting	4-26
:FILE:FORMat?	Queries all the settings relating to floppy disk formatting	4-26

Command	Description	Page
:FILE:FORMat:EXECute	Formats the floppy disk	4-26
:FILE:FORMat:TYPE	Selects the floppy disk format to be used for formatting, or queries the current setting	4-26
:FILE:INFormation	Checks floppy disk file information	4-26
:FILE:LOAD?	Queries the settings relating to loading data	4-26
:FILE:LOAD:ACQuisition:EXECute	Executes the loading of acquisition waveform data	4-26
:FILE:LOAD:ACQuisition:NAME	Sets the name of the acquisition waveform data file to be loaded	4-26
:FILE:LOAD:SETup:EXECute	Loads setting parameters	4-26
:FILE:LOAD:SETup:NAME	Sets the name of the setting parameter file to be loaded	4-26
:FILE:LOAD:TRACe?	Queries all the setting values relating to loading P-P waveform data	4-26
:FILE:LOAD:TRACe:EXECute	Loads P-P waveform data	4-26
:FILE:LOAD:TRACe:NAME	Sets the name of the P-P waveform data file to be loaded	4-26
:FILE:LOAD:TRACe:TRACe	Sets the trace No. where the P-P waveform data will be loaded	4-26
:FILE:PROTeCt:ACQuisition:EXECute	Protects acquisition data	4-26
:FILE:PROTeCt:ACQuisition:NAME	Sets the name of the acquisition waveform data file to be protected	4-26
:FILE:PROTeCt:SETup:EXECute	Protects setting parameters	4-27
:FILE:PROTeCt:SETup:NAME	Sets the name of the setting parameter file to be protected	4-27
:FILE:PROTeCt:TRACe:EXECute	Protects P-P waveform data	4-27
:FILE:PROTeCt:TRACe:NAME	Sets the name of the P-P waveform data file to be protected	4-27
:FILE:SAVE?	Queries all settings relating to saving data	4-27
:FILE:SAVE:ACQuisition:EXECute	Saves acquisition waveform data	4-27
:FILE:SAVE:ACQuisition:NAME	Sets the name of the acquisition waveform data file to be saved	4-27
:FILE:SAVE:SETup:EXECute	Saves setting parameters	4-27
:FILE:SAVE:SETup:NAME	Sets the name of the setting parameter file to be saved	4-27
:FILE:SAVE:TRACe?	Queries all settings relating to saving P-P waveform data	4-27
:FILE:SAVE:TRACe:EXECute	Saves P-P waveform data	4-27
:FILE:SAVE:TRACe:NAME	Sets the name of the P-P waveform data file to be saved	4-27
:FILE:SAVE:TRACe:TRACe	Selects the trace No. of the P-P waveform to be saved, or queries the current setting	4-27
:FILE:SAVE:ACQuisition:FORM	Sets data format for waveform data save, or queries current setting	4-27
<b>FILTer Group</b>		
:FILTer?	Queries all the settings relating to the input filter	4-28
:FILTer:BANdwidth	Sets the bandwidth limits, or queries the current setting	4-28
:FILTer:SMOothing	Turns smoothing ON/OFF, or queries the current setting	4-28
<b>GONogo Group</b>		
:GONogo?	Queries all the GO/NO-GO settings	4-31
:GONogo[:MODE]	Turns GO/NO-GO determination ON/OFF, or queries the current setting	4-31
:GONogo:PARAmeter?	Queries all the settings relating to GO/NO-GO determination which use parameters	4-31
:GONogo:PARAmeter:ABORt	Aborts parameter GO/NO-GO determination	4-31
:GONogo:PARAmeter:ACTion	Sets the action to be taken if the parameter determination result is NO-GO, or queries the current setting	4-31
:GONogo:PARAmeter:BUZZer	Sets the ON/OFF setting of the buzzer when NO-GO occurs during parameter GO/NO-GO determination, or queries the current setting	4-31
:GONogo:PARAmeter:COUNt?	Queries the number of times parameter GO/NO-GO is performed	4-31
:GONogo:PARAmeter:EXECute	Executes parameter GO/NO-GO determination	4-31
:GONogo:PARAmeter:FNAME	Selects the file name of the NG data to be output	4-31
:GONogo:PARAmeter:ITEM<x>?	Queries all the settings relating to the specified waveform parameter for parameter GO/NO-GO determination	4-31
:GONogo:PARAmeter:ITEM<x>:CAUSe?	Queries whether the specified waveform parameter used for parameter GO/NO-GO determination is the cause of failure	4-32
:GONogo:PARAmeter:ITEM<x>:CONDition	Sets the criteria for the specified waveform parameter used for parameter GO/NO-GO determination, or queries the current setting	4-32
:GONogo:PARAmeter:ITEM<x>:TRACe	Sets the target waveform for the specified waveform parameter used for parameter GO/NO-GO determination, or queries the current setting	4-32
:GONogo:PARAmeter:ITEM<x>:TYPE?	Sets the upper/lower limits for the specified parameter used for parameter GO/NO-GO determination, or queries the current setting	4-32
:GONogo:PARAmeter:ITEM<x>:TYPE:<parameter>	Sets the upper/lower limits for the specified measurement item, or queries the current setting	4-32

## 4.1 List of Commands

Command	Description	Page
:GONogo:PARAmeter:ITEM<x>:VALue?	Queries the measured value of the specified parameter used for parameter GO/NO-GO determination	4-32
:GONogo:PARAmeter:LOGic	Sets the determination logic for parameter GO/NO-GO, or queries the current setting	4-32
:GONogo:PARAmeter:NGCount?	Queries the number of NGs occurring during parameter GO/NO-GO determination	4-32
:GONogo:PARAmeter:SEQuence	Sets the number of times parameter GO/NO-GO determination is performed, or queries the current setting	4-33
:GONogo:PARAmeter:STOPcount	Sets the number of times to execute parameter determination, or queries the current setting	4-33
:GONogo:PARAmeter:TRANge	Sets the parameter GO/NO-GO determination range, or queries the current setting	4-33
:GONogo:PARAmeter:WAIT?	Wait for parameter determination to finish	4-33
:GONogo:TYPE	Sets the GO/NO-GO type, or queries the current setting	4-33
:GONogo:ZONE?	Queries all the settings relating to zone GO/NO-GO determination	4-33
:GONogo:ZONE:ABORt	Aborts GO/NO-GO determination	4-33
:GONogo:ZONE:ACTion	Sets the action to be taken if the zone determination is NG, or queries the current setting	4-33
:GONogo:ZONE:BUZZer	Sets the ON/OFF setting of the buzzer when NO-GO occurs during zone GO/NO-GO determination, or queries the current setting	4-33
:GONogo:ZONE:COUNT?	Queries the number of times zone GO/NO-GO determination is performed	4-33
:GONogo:ZONE:EDIT<x>:EXIT	Exits from the edit menu for the specified zone	4-34
:GONogo:ZONE:EDIT<x>:PART	Selects partial edit for the specified zone during zone GO/NO-GO determination	4-34
:GONogo:ZONE:EDIT<x>:WHOLe	Selects total edit for the specified zone	4-34
:GONogo:ZONE:EXECute	Executes zone determination	4-34
:GONogo:ZONE:FNAME	Selects the file name of the NO-GO data to be output	4-34
:GONogo:ZONE:LOGic	Sets the determination logic for zone GO/NO-GO determination, or queries the current settings	4-34
:GONogo:ZONE:NGCount?	Queries the number of NGs occurring during zone GO/NO-GO determination	4-34
:GONogo:ZONE:SEQuence	Sets the number of times GO/NO-GO determination is performed, or queries the current setting	4-34
:GONogo:ZONE:STOPcount	Sets the number of times to execute zone determination, or queries the current setting	4-34
:GONogo:ZONE:TRACe<x>?	Queries all the settings made for the specified zone for zone GO/NO-GO determination	4-34
:GONogo:ZONE:TRACe<x>:CAUSE?	Queries whether the specified trace used for zone GO/NO-GO determination is the cause of failure	4-34
:GONogo:ZONE:TRACe<x>:CONDition	Sets the criteria for the specified trace used for zone GO/NO-GO determination, or queries the current setting	4-35
:GONogo:ZONE:TRACe<x>:MODE	Selects whether the zone determination result for the specified trace is validated or not, or queries the current setting	4-35
:GONogo:ZONE:TRACe<x>:ZONE	Sets the zone data for each trace, or queries the current setting	4-35
:GONogo:ZONE:WAIT?	Wait for zone determination to finish	4-35
<b>HCOPy Group</b>		
:HCOPy?	Queries all the settings relating to output of screen data	4-39
:HCOPy:COMMeNt?	Queries all the settings relating to the comment to be printed	4-39
:HCOPy:COMMeNt:CONTeNts	Sets the comment to be printed	4-39
:HCOPy:COMMeNt:MODE	Turns the printing of comments ON/OFF, or queries the current setting	4-39
:HCOPy:EQUipment	Selects the device to which the screen data is to be output, or queries the current setting	4-39
:HCOPy:EXTPriNter?	Queries all the settings relating to output an external printer	4-40
:HCOPy:EXTPriNter:FORMat	Sets the data format for printout to an external printer, or queries the current setting	4-40
:HCOPy:EXTPriNter:TONE	Sets the half tone of the external printer output, or queries the current setting	4-40
:HCOPy:FLOPPy?	Queries all the settings relating to saving of screen image data on a floppy disk	4-40
:HCOPy:FLOPPy:ANAMing	Sets the auto naming mode for the file name of the screen image data to be output, or queries the current setting	4-40
:HCOPy:FLOPPy:COMPreSSion	Set the compression of the screen image data to output ON/OFF, or queries the current setting	4-40
:HCOPy:FLOPPy:EXECute	Saves the screen image data	4-40
:HCOPy:FLOPPy:FORMat	Selects the format of the screen image data to be saved, or queries the current setting	4-40
:HCOPy:FLOPPy:NAME	Selects the name of the screen image data file to be output	4-40
:HCOPy:FLOPPy:PEN?	Queries all the settings relating to the pens of the image data to be output	4-40

Command	Description	Page
:HCOPY:FLOPY:PEN:AUTO	Sets the number of pens of the image data to be output in case pens are assigned automatically, or queries the current setting	4-40
:HCOPY:FLOPY:PEN:MANual?	Queries the number of pens of the image data to be output in case pens are assigned automatically	4-40
:HCOPY:FLOPY:PEN:MANual:GRID	Sets the pen Nos. for the graticule etc. in case pens are assigned manually, or queries the current setting	4-40
:HCOPY:FLOPY:PEN:MANual:LOAD<x>	Sets the pen Nos. of all loaded traces in case pens are assigned manually, or queries the current setting	4-41
:HCOPY:FLOPY:PEN:MANual:TRACe<x>	Sets the pen No. for the specified trace in case pens are assigned manually, or queries the current setting	4-41
:HCOPY:FLOPY:PEN:MODE	Sets the pen assignment method, or queries the current setting	4-41
:HCOPY:FLOPY:PFORmat	Sets the output size for the screen image data, or queries the current setting	4-41
:HCOPY:FLOPY:SPEed	Sets the pen speed of the screen image data to be output, or queries the current setting	4-41
:HCOPY:FLOPY:TONE	Set the color tone of the screen image data to output, or queries the current setting	4-41
:HCOPY:FLOPY:XYMode	Selects the plot method for X-Y waveforms, or queries the current setting	4-41
:HCOPY:PLOTter?	Queries all the settings relating to output to an HP-GL plotter	4-41
:HCOPY:PLOTter:PEN?	Queries all the settings relating to the plotter's pen	4-41
:HCOPY:PLOTter:PEN:AUTO	Sets the number of pens in case pens are assigned automatically, or queries the current setting	4-41
:HCOPY:PLOTter:PEN:MANual?	Queries the number of pens in case pens are assigned manually	4-41
:HCOPY:PLOTter:PEN:MANual:GRID	Sets the pen Nos. for the graticule etc. in case pens are assigned manually, or queries the current setting	4-41
:HCOPY:PLOTter:PEN:MANual:LOAD<x>	Sets the plotter pen No. for the specified loaded trace in case pens are assigned manually, or queries the current setting	4-42
:HCOPY:PLOTter:PEN:MANual:TRACe<x>	Sets the plotter pen No. for the specified loaded trace in case pens are assigned manually, or queries the current setting	4-42
:HCOPY:PLOTter:PEN:MODE	Sets the plotter pen assignment method, or queries the current setting	4-42
:HCOPY:PLOTter:PFORmat	Sets the output size to the plotter, or queries the current setting	4-42
:HCOPY:PLOTter:SPEed	Sets the plotter pen speed, or queries the current setting	4-42
:HCOPY:PLOTter:XYMode	Selects the plot method for X-Y waveforms, or queries the current setting	4-42
:HCOPY:PRINter?	Queries all the settings relating to output to the built-in printer	4-42
:HCOPY:PRINter:ABORt	Aborts print-out to the built-in printer	4-42
:HCOPY:PRINter:EXECute	Prints out to the built-in printer	4-42
:HCOPY:PRINter:POSition	Sets the position of the expansion box when printing out a long copy, or queries the current setting	4-42
:HCOPY:PRINter:RANGe	Sets the output range for a long copy, or queries the current setting	4-42
:HCOPY:PRINter:REPORt	Selects whether a report is to be output, or queries the current setting	4-42
:HCOPY:PRINter:TYPE	Selects short or long copy, or queries the current setting	4-43
:HCOPY:PRINter:ZOOM	Sets the zoom rate used for a long copy, or queries the current setting	4-43
:HCOPY:SCSI?	Queries all the settings relating to saving of screen image data on a SCSI device	4-43
:HCOPY:SCSI:ANAMing	Sets the auto naming mode for the file name of the screen image data to be output, or queries the current setting	4-43
:HCOPY:SCSI:COMPReSSion	Set the compression of the screen image data to output ON/OFF, or queries the current setting	4-43
:HCOPY:SCSI:EXECute	Saves the screen image data to SCSI	4-43
:HCOPY:SCSI:FORMat	Selects the format of the screen image data to be saved, or queries the current setting	4-43
:HCOPY:SCSI:NAME	Selects the name of the screen image data file to be output	4-43
:HCOPY:SCSI:PEN?	Queries all the settings relating to the pens of the image data to be output	4-43
:HCOPY:SCSI:PEN:AUTO	Sets the number of pens of the image data to be output when pens are assigned automatically, or queries the current setting	4-43
:HCOPY:SCSI:PEN:MANual?	Queries all the settings relating the number of pens of the image data to be output when pens are assigned manually	4-44
:HCOPY:SCSI:PEN:MANual:GRID	Sets the pen Nos. for the graticule, etc. when pens are assigned manually, or queries the current setting	4-44
:HCOPY:SCSI:PEN:MANual:LOAD<x>	Sets the pen Nos. of all loaded traces when pens are assigned manually, or queries the current setting	4-44
:HCOPY:SCSI:PEN:MANual:TRACe<x>	Sets the pen No. for the specified trace when pens are assigned manually, or queries the current setting	4-44

## 4.1 List of Commands

Command	Description	Page
:HCOPY:SCSI:PEN:MODE	Sets the pen assignment method, or queries the current setting	4-44
:HCOPY:SCSI:PFORmat	Sets the output size for the screen image data, or queries the current setting	4-44
:HCOPY:SCSI:SPEEd	Sets the pen speed, or queries the current setting	4-44
:HCOPY:SCSI:TONE	Set the color tone of the screen image data to output, or queries the current setting	4-44
:HCOPY:SCSI:XYMode	Selects the plot method for X-Y waveforms, or queries the current setting	4-44
<b>IMAGe Group</b>		
:IMAGe?	Queries all screen-image data output settings	4-45
:IMAGe:FORMat	Sets/queries screen-image data output format	4-45
:IMAGe:SEND?	Queries screen-image data	4-45
:IMAGe:TONE	Set the color tone of the screen image data to output, or queries the current setting	4-45
<b>INITialize Group</b>		
:INITialize:EXECute	Executes initialization	4-45
<b>MATH Group</b>		
:MATH?	Queries all the settings relating to waveform computation	4-46
:MATH:FFT?	Queries all the settings relating to FFT computation	4-46
:MATH:FFT:DISPLay	Sets the FFT waveform to be displayed, or queries the current setting	4-46
:MATH:FFT:MODE	Sets the FFT waveform display ON/OFF, or queries the current setting	4-46
:MATH:FFT:TRACe	Sets the trace for FFT measurement, or queries the current setting	4-47
:MATH:FFT:WINDow	Sets the FFT window function, or queries the current setting	4-47
:MATH:OPERator	Sets the computation contents	4-47
:MATH:PHASe?	Queries all the settings relating to phase shifts	4-47
:MATH:PHASe[:SHIFt]	Sets the desired phase shift, or queries the current setting	4-47
:MATH:SENSitivity	Sets the display sensitivity of the vertical axis, or queries the current setting	4-47
<b>MEASure Group</b>		
:MEASure?	Queries all the settings relating to automatic measurement of waveform parameters	4-50
:MEASure:ALLScan:ABORt	Aborts automatic measurement of acquisition data	4-50
:MEASure:ALLScan:EXECute	Starts automatic measurement of acquisition data	4-50
:MEASure:{LOAD<x> TRACe<x>}?	Queries all the settings relating to the measurement items for each waveform	4-51
:MEASure:{LOAD<x> TRACe<x>}:ALL	Sets the parameters for the specified waveform ON/OFF	4-51
:MEASure:{LOAD<x> TRACe<x>}:COPY	Copies the ON/OFF settings of all measurement items of a specific channel to all other waveforms	4-51
:MEASure:{LOAD<x> TRACe<x>}:DELay?	Queries all the settings relating to delay measurement between channels for a specified waveform	4-51
:MEASure:{LOAD<x> TRACe<x>}:DELay:MEASure?	Queries all the settings relating to the waveform which is the subject of delay measurement between channels for a specified waveform	4-51
:MEASure:{LOAD<x> TRACe<x>}:DELay:MEASure:COUNT	Sets the count number of the waveform which is the subject of delay measurement, or queries the current setting	4-51
:MEASure:{LOAD<x> TRACe<x>}:DELay:MEASure:LEVeL?	Queries all the settings relating to the threshold level	4-51
:MEASure:{LOAD<x> TRACe<x>}:DELay:MEASure:LEVeL:PERCent	Sets the % value of the threshold level, or queries the current setting	4-51
:MEASure:{LOAD<x> TRACe<x>}:DELay:MEASure:LEVeL:UNIT	Sets the voltage value of the threshold level, or queries the current setting	4-52
:MEASure:{LOAD<x> TRACe<x>}:DELay:MEASure:SLOPe	Sets the slope of the waveform which is the subject of delay measurement between channels for a specified waveform, or queries the current setting	4-52
:MEASure:{LOAD<x> TRACe<x>}:DELay[:MODE]	Sets the delay measurement between channels for a specified waveform ON/OFF, or queries the current setting	4-52
:MEASure:{LOAD<x> TRACe<x>}:DELay:REFerence?	Queries all settings relating to the reference waveform of delay measurement between channels for a specified waveform	4-52
:MEASure:{LOAD<x> TRACe<x>}:DELay:REFerence:COUNT	Sets the count number of the reference waveform of delay measurement between channels for a specified waveform, or queries the current setting	4-52

Command	Description	Page
:MEASure:{LOAD<x> TRACe<x>}:DELay:REFerence:LEVel?	Queries all the settings relating to the threshold level of the reference waveform of delay measurement between channels for a specified waveform	4-52
:MEASure:{LOAD<x> TRACe<x>}:DELay:REFerence:LEVel:PERCent	Sets the % value of the threshold level of the reference waveform of delay measurement between channels for a specified waveform, or queries the current setting	4-52
:MEASure:{LOAD<x> TRACe<x>}:DELay:REFerence:LEVel:UNIT	Set the voltage value of the threshold level of the reference waveform of delay measurement of delay measurement between channels for a specified waveform, or queries the current setting	4-53
:MEASure:{LOAD<x> TRACe<x>}:DELay:REFerence:SLOPe	Sets the slope of the reference waveform, or queries the current setting	4-53
:MEASure:{LOAD<x> TRACe<x>}:DELay:REFerence:TRACe	Sets the trace of the reference waveform, or queries the current setting	4-53
:MEASure:{LOAD<x> TRACe<x>}:DPRoximal?	Queries all settings relating to distal, mesial, and proximal points	4-53
:MEASure:{LOAD<x> TRACe<x>}:DPRoximal:PERCent	Sets the distal, mesial, and proximal points in percentages, or queries the current settings	4-53
:MEASure:{LOAD<x> TRACe<x>}:DPRoximal:UNIT	Sets the distal, mesial, and proximal points in terms of voltages, or queries the current settings	4-53
:MEASure[:MODE]	Turns automated measurements ON/OFF, or queries the current setting	4-53
:MEASure:{LOAD<x> TRACe<x>}:<parameter>	Turns a measurement item ON/OFF for a specified waveform, or queries the current setting	4-53
:MEASure:TRANge	Sets the measurement range, or queries the current setting	4-54
:MEASure:{LOAD<x> TRACe<x>}:VALue? <parameter>	Queries the automatically measured value of the specified measurement item for the specified waveform	4-54
:MEASure:WAIT?	Wait for automatic measurement to finish	4-54
<b>RECall Group</b>		
:RECall?	Queries all settings relating to the recalling of data from the internal memory	4-55
:RECall:LOAD<x>	Sets the display of the loaded waveform ON/OFF, or queries the current setting	4-55
:RECall:MEMory<x>	Recalls waveform data from the specified internal memory and recalls it as a specified trace	4-55
:RECall:POSition<x>	Sets the vertical position of the recalled waveform, or queries the current setting	4-55
:RECall:SETup	Recalls the specified set-up data from the internal memory	4-55
<b>SCAlE Group</b>		
:SCAlE?	Queries all the settings relating to linear scaling	4-56
:SCAlE:CHANnel<x>?	Queries all the settings relating to linear scaling for a specified channel	4-56
:SCAlE:CHANnel<x>:A	Sets the linear scaling A value for a specified channel, or queries the current setting	4-57
:SCAlE:CHANnel<x>:B	Sets the linear scaling B value for a specified channel, or queries the current setting	4-57
:SCAlE:CHANnel<x>:MODE	Sets linear scaling ON/OFF, or queries the current setting	4-57
:SCAlE:CHANnel<x>:UNIT	Sets the linear scaling unit for a specific channel, or queries the current setting	4-57
:SCAlE:LOAD<x>?	Queries all the settings relating to linear scaling for a specified loaded waveform	4-57
:SCAlE:LOAD<x>:A	Sets the linear scaling A value for a specified loaded waveform, or queries the current setting	4-57
:SCAlE:LOAD<x>:B	Sets the linear scaling B value for a specified loaded waveform, or queries the current setting	4-57
:SCAlE:LOAD<x>:MODE	Sets linear scaling for a specified loaded waveform ON/OFF, or queries the current setting	4-57
:SCAlE:LOAD<x>:UNIT	Sets the linear scaling unit for a specified loaded waveform, or queries the current setting	4-57
:SCAlE:MODE	Sets linear scaling ON/OFF, or queries the current setting	4-57
<b>SNAP Group</b>		
:SNAP	Executes a snapshot	4-58



## 4.1 List of Commands

Command	Description	Page
<b>START Group</b>		
:START	Starts waveform acquisition	4-58
<b>STATus Group</b>		
:STATus?	Queries all the settings relating to the status of the communication function	4-59
:STATus:CONDition?	Queries the contents of the condition register	4-59
:STATus:EESE	Sets the extended event enable register, or queries the current setting	4-59
:STATus:EESR?	Queries the contents of the extended event register and clears the register	4-59
:STATus:ERRor?	Queries the code and the message of the error which has occurred	4-59
:STATus:FILTer<x>	Queries all the settings relating to the specified transition filter, or queries the current setting	4-59
:STATus:QENable	Selects whether or not to store a message other than an error message into the error queue, or queries the current setting	4-59
:STATus:QMESsage	Selects whether or not to add the message contents to a response to "STATus:ERRor?", or queries the current setting	4-59
:STATus:SPOLL?	Executes a serial poll	4-59
<b>STOP Group</b>		
:STOP	Stops waveform acquisition	4-59
<b>STORe Group</b>		
:STORe:SETup	Stores setting parameters in the internal memory	4-60
:STORe:TRACe<x>	Stores waveform data in the internal memory	4-60
<b>SYSTem Group</b>		
:SYSTem?	Queries all the system settings	4-61
:SYSTem:BATTery?	Queries the condition of the internal lithium battery	4-61
:SYSTem:CLOCK?	Queries all the settings relating to the date and time	4-61
:SYSTem:CLOCK:DATE	Sets the date, or queries the current setting	4-61
:SYSTem:CLOCK:MODE	Turns display of the date and time ON/OFF, or queries the current setting	4-61
:SYSTem:CLOCK:TIME	Sets the time, or queries the current setting	4-61
:SYSTem:CONFig?	Queries all the settings relating to the configuration system	4-61
:SYSTem:CONFig:PDENsity	Selects the level of the printing density, or queries the current setting	4-61
:SYSTem:LCD?	Queries all settings relating to the LCD	4-61
:SYSTem:LCD:BRIGhtness	Sets the brightness of the LCD, or queries the setting	4-61
:SYSTem:LCD:MODE	Turns the back light of the LCD ON/OFF, or queries the current setting	4-61
<b>TIMEbase Group</b>		
:TIMEbase?	Queries all the timebase settings	4-62
:TIMEbase:SOURce	Sets the timebase, or queries the current setting	4-62
:TIMEbase:SRATE?	Queries the current sample rate	4-62
:TIMEbase:TDIV	Set the time axis, or queries the current setting	4-62
<b>TRIGger Group</b>		
:TRIGger?	Queries all trigger settings	4-65
:TRIGger:ACTion?	Queries all the settings relating to action-on-trigger	4-65
:TRIGger:ACTion:FNAME	Sets the file name when saving upon an action-on-trigger	4-65
:TRIGger:ACTion:MODE	Selects the action-on-trigger mode, or queries the current setting	4-65
:TRIGger:ACTion:START	Starts the action-on-trigger	4-66
:TRIGger:ACTion:STOP	Stops the action-on-trigger	4-66
:TRIGger:MODE	Sets the trigger mode, or queries the current setting	4-66
:TRIGger:POSition	Sets the trigger position, or queries the current setting	4-66
:TRIGger:SIMPLe?	Queries all trigger settings	4-66
:TRIGger[:SIMPLe]:DELay	Sets the delay time for trigger, or queries the current setting	4-66
:TRIGger[:SIMPLe]:EDGE?	Queries all the settings when an edge trigger is selected	4-66
:TRIGger[:SIMPLe]:EDGE:SLOPe	Sets the trigger slope in case of an edge trigger, or queries the current setting	4-66
:TRIGger[:SIMPLe]:EDGE:SOURce	Sets the trigger source in case of an edge trigger, or queries the current setting	4-66
:TRIGger[:SIMPLe]:FUNCTion	Sets the type of trigger, or queries the current setting	4-66
:TRIGger[:SIMPLe]:HOLDoff?	Queries all the holdoff settings for a trigger	4-66
:TRIGger[:SIMPLe]:HOLDoff:MODE	Turns holdoff ON/OFF for a trigger, or queries the current setting	4-67
:TRIGger[:SIMPLe]:HOLDoff:TIME	Sets the holdoff time for a trigger, or queries the current setting	4-67
:TRIGger[:SIMPLe]:OR?	Queries all the settings when an OR trigger is selected	4-67

Command	Description	Page
:TRIGger[:SIMPlE]:OR:CHANnel<x>	Setst the slope and state when an OR trigger is selected, or queries the current setting	4-67
:TRIGger[:SIMPlE]:PATTern?	Queries all the settings when a pattern trigger is selected	4-67
:TRIGger[:SIMPlE]:PATTern:CHANnel<x>	Sets the state for the specified channel for a pattern trigger, or queries the current setting	4-67
:TRIGger[:SIMPlE]:PATTern:CLOCK?	Queries all the clock settings when a pattern trigger is selected	4-67
:TRIGger[:SIMPlE]:PATTern:CLOCK:SLOPe	Sets the slope for the clock channel for a pattern trigger, or queries the current setting	4-67
:TRIGger[:SIMPlE]:PATTern:CLOCK:SOURce	Sets the clock channel for a pattern trigger, or queries the current setting	4-67
:TRIGger[:SIMPlE]:PATTern:CONDition	Sets the trigger condition for a pattern trigger, or queries the current setting	4-67
:TRIGger[:SIMPlE]:TV?	Queries all the settings when a TV trigger has been selected	4-67
:TRIGger[:SIMPlE]:TV:FIELD	Sets the field No. for a TV trigger, or queries the current setting	4-68
:TRIGger[:SIMPlE]:TV:LEVeL	Sets the trigger level, or queries the current setting	4-68
:TRIGger[:SIMPlE]:TV:LINE	Sets the line No., or queries the current setting	4-68
:TRIGger[:SIMPlE]:TV:POLarity	Sets the polarity, or queries the current setting	4-68
:TRIGger[:SIMPlE]:TV:TYPE	Sets the broadcasting system, or queries the current setting	4-68
:TRIGger[:SIMPlE]:WIDTH?	Queries all the settings when a pulse trigger is selected	4-68
:TRIGger[:SIMPlE]:WIDTH:CONDition	Sets the trigger condition for a pulse trigger, or queries the current setting	4-68
:TRIGger[:SIMPlE]:WIDTH:SLOPe	Sets the trigger state for a pulse trigger, or queries the current setting	4-68
:TRIGger[:SIMPlE]:WIDTH:SOURce	Sets the trigger source for a pulse trigger, or queries the current setting	4-68
:TRIGger[:SIMPlE]:WIDTH:TIME	Sets the pulse width for a pulse trigger, or queries the current setting	4-68
:TRIGger[:SIMPlE]:WINDow?	Queries all settings relating to the window trigger	4-68
:TRIGger[:SIMPlE]:WINDow:CONDition	Sets the trigger condition for a window trigger, or queries the current setting	4-69
:TRIGger[:SIMPlE]:WINDow:LEVeL	Sets the center level for a window trigger, or queries the current setting	4-69
:TRIGger[:SIMPlE]:WINDow:WIDTH	Sets the width of the window, or queries the current setting	4-69
:TRIGger:SOURce?	Queries all the settings relating to the trigger source	4-69
:TRIGger:SOURce:CHANnel<x>?	Queries all the trigger source settings for the specified channel	4-69
:TRIGger:SOURce:CHANnel<x>:LEVeL	Sets the trigger level, or queries the current setting	4-69
:TRIGger:SOURce:COUPling	Sets the trigger coupling, or queries the current setting	4-69
:TRIGger:SOURce:EXTernal?	Queries the external trigger settings	4-69
:TRIGger:SOURce:EXTernal:LEVeL	Sets the external trigger level, or queries the current setting	4-69
:TRIGger:SOURce:HFRejection	Turns the HF rejection ON/OFF, or queries the current setting	4-69
<b>UNIT Group</b>		
:UNIT?	Queries the default values of all units	4-70
:UNIT:FREQuency	Sets the default value for <Frequency>, or queries the current setting	4-70
:UNIT:TIME	Sets the default value for <Time>, or queries the current setting	4-70
:UNIT:VOLTage	Sets the default value for <Voltage>, or queries the current setting	4-70
<b>WAVeform Group</b>		
:WAVeform?	Queries all information relating to the waveform data	4-71
:WAVeform:BITS?	Queries the bit length of the waveform data	4-71
:WAVeform:BYTeorder	Sets the transmission order when words consist of more than 2 bytes, or queries the current setting	4-71
:WAVeform:DATaselect	Selects whether acquisition data is ACQ waveform data or P-P waveform data, or queries the current setting	4-71
:WAVeform:END	Sets the point at which the last item is to be located, or queries the current setting	4-72
:WAVeform:FORMat	Selects the format in which the data is to be formatted, or queries the current setting	4-72
:WAVeform:LENGth?	Queries the number of records in the waveform	4-72
:WAVeform:OFFSet?	Queries the offset voltage of the waveform	4-72
:WAVeform:POSition?	Queries the vertical-axis position	4-72
:WAVeform:SEND?	Queries the waveform data	4-72
:WAVeform:SIGN?	Queries whether the data is signed or unsigned when binary data is used	4-72
:WAVeform:SRATe?	Queries the sample rate of the waveform	4-72
:WAVeform:STARt	Sets the point at which the first item is to be located, or queries the current setting	4-73

## 4.1 List of Commands

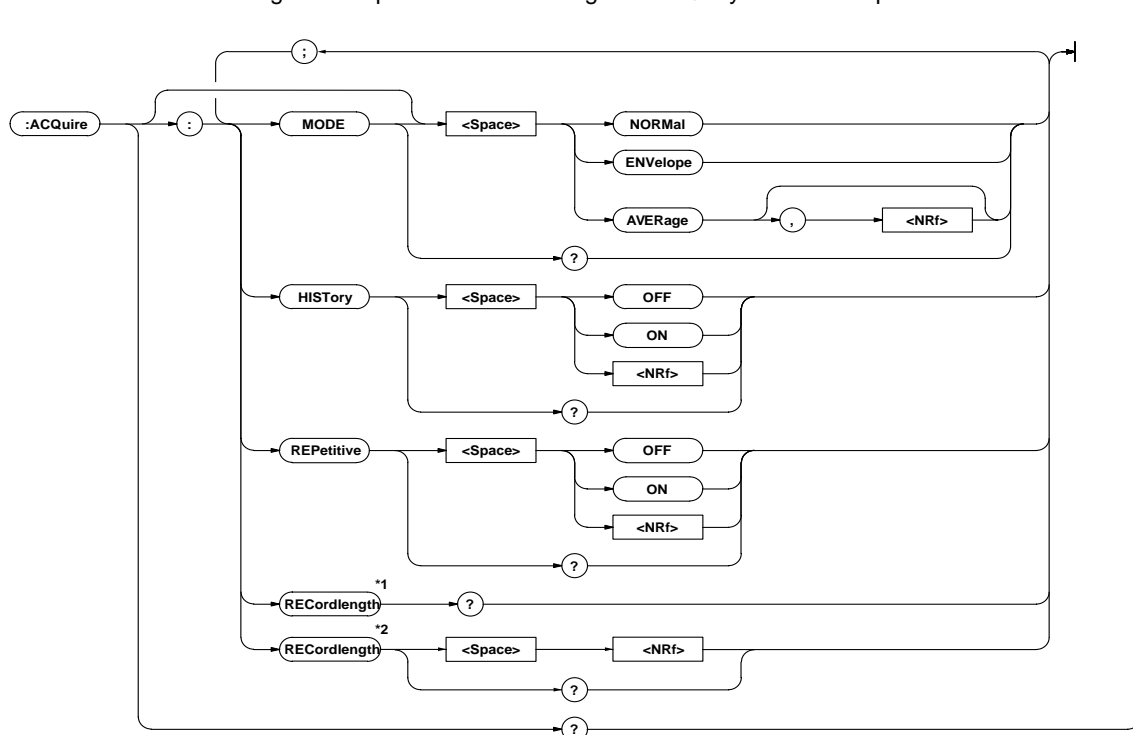
---

Command	Description	Page
:WAVeform:TRACe	Sets the target waveform, or queries the current setting	4-73
:WAVeform:TRIGger?	Queries the points elapsed from the start point until the trigger point	4-73
:WAVeform:TYPE?	Queries the type of ACQ waveform data	4-73
:WAVeform:VDIV?	Queries the V/div value	4-73
<b>ZOOM Group</b>		
:ZOOM?	Queries all the settings relating to waveform expansion	4-73
:ZOOM:HPOSition	Sets the center position for time axis expansion, or queries the current setting	4-73
:ZOOM:HZOm	Sets the expansion rate in the horizontal direction, or queries the current setting	4-73
:ZOOM[:MODE]	Sets the display type for the expanded waveform, or queries the current setting	4-73
<b>Common Command Group</b>		
*CAL?	Performs calibration and queries the result	4-74
*CLS	Clears the standard event register, extended event register and error queue	4-74
*ESE	Sets the value for the standard event enable register, or queries the current setting	4-74
*ESR?	Queries the value of the standard event register and clears it at the same time	4-75
*IDN?	Queries the instrument model	4-75
*OPC	Sets the OPC event after execution of the overlap command	4-75
*OPC?	Queries whether the overlap command has been executed	4-75
*OPT?	Queries installed options	4-75
*PSC	Selects whether or not to clear registers when power is turned ON, or queries the current setting	4-75
*RST	Resets the current settings	4-75
*SRE	Sets the value of the service request enable register, or queries the current setting	4-75
*STB?	Queries the value of the status byte register	4-76
*TST?	Executes a self-test and queries the result	4-76
*WAI	Waits for the command following “*WAI” until execution of the designated overlap command has been completed	4-76

---

## 4.2 ACQUIRE Group

The commands in the ACQ group are used to make settings relating to, and inquires about acquisition. This allows you to make the same settings and inquiries as when using the ACQ key on the front panel.



\*1 Available on the DL1540C

\*2 Available on DL1540CL

### ACQUIRE?

**Function** Queries all the acquisition settings.

**Syntax** ACQUIRE?

**Example** ACQUIRE?→:ACQUIRE:MODE NORMAL;  
REPETITIVE 0;RECORDLENGTH 10000;HISTORY 0

### ACQUIRE:HISTORY

**Function** Sets history memory function ON/OFF, or queries the current setting.

**Syntax** ACQUIRE:HISTORY {<Boolean>}  
ACQUIRE:HISTORY?

**Example** ACQUIRE:HISTORY ON  
ACQUIRE:HISTORY?→:ACQUIRE:HISTORY 1

### ACQUIRE[:MODE]

**Function** Sets acquisition mode, or queries the current setting.

**Syntax** ACQUIRE[:MODE] {AVERAGE[, {<NRf>}]  
ENVELOPE|NORMAL}  
ACQUIRE:MODE?  
{<NRf>}=2 to 256(in steps of 2<sup>n</sup>)

**Example** ACQUIRE:MODE NORMAL  
ACQUIRE:MODE?→:ACQUIRE:MODE NORMAL

**Description** It is also possible to set the averaging count if averaging mode has been selected as the acquisition mode.

### ACQUIRE:RECORDLENGTH?

**Function** Queries the record length.

**Syntax** ACQUIRE:RECORDLENGTH?

**Example** ACQUIRE:RECORDLENGTH?→:ACQUIRE:  
RECORDLENGTH 10020

**Description** Available on the DL1540C.

### ACQUIRE:RECORDLENGTH

**Function** Sets the record length, or queries the current setting.

**Syntax** ACQUIRE:RECORDLENGTH {<NRf>}  
ACQUIRE:RECORDLENGTH?  
<NRf>=1000,10000,100000,  
400000,1000000,2000000

**Example** ACQUIRE:RECORDLENGTH 10000  
ACQUIRE:RECORDLENGTH?→:ACQUIRE:  
RECORDLENGTH 10000

**Description** Available on DL1540CL.

### ACQUIRE:REPETITIVE

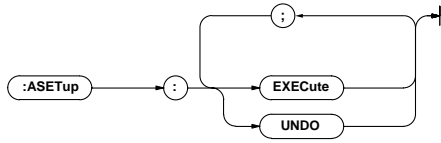
**Function** Turns repetitive sampling mode ON/OFF, or queries the current setting.

**Syntax** ACQUIRE:REPETITIVE {<Boolean>}  
ACQUIRE:REPETITIVE?

**Example** ACQUIRE:REPETITIVE OFF  
ACQUIRE:REPETITIVE?→:ACQUIRE:  
REPETITIVE 0

### 4.3 ASETup Group

The commands in the ASETup group are used to make settings relating to, and inquiries about auto set-up. This allows you to make the same settings as when using the AUTO-SETUP key on the front panel.



#### ASETup:EXECute

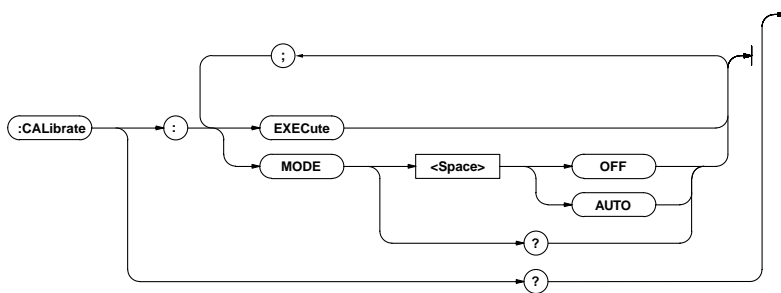
**Function** Executes auto set-up.  
**Syntax** ASETup:EXECute  
**Example** ASETUP:EXECUTE

#### ASETup:UNDO

**Function** Cancels auto set-up settings.  
**Syntax** ASETup:UNDO  
**Example** ASETUP:UNDO

### 4.4 CALibrate Group

The commands in the CALibrate group are used to make settings relating to, or inquiries about calibration. This allows you to make the same settings and inquiries as when using the CAL key on the front panel.



#### CALibrate?

**Function** Queries all the calibration settings.  
**Syntax** CALibrate?  
**Example** CALIBRATE?→:CALIBRATE:MODE AUTO

#### CALibrate:EXECute

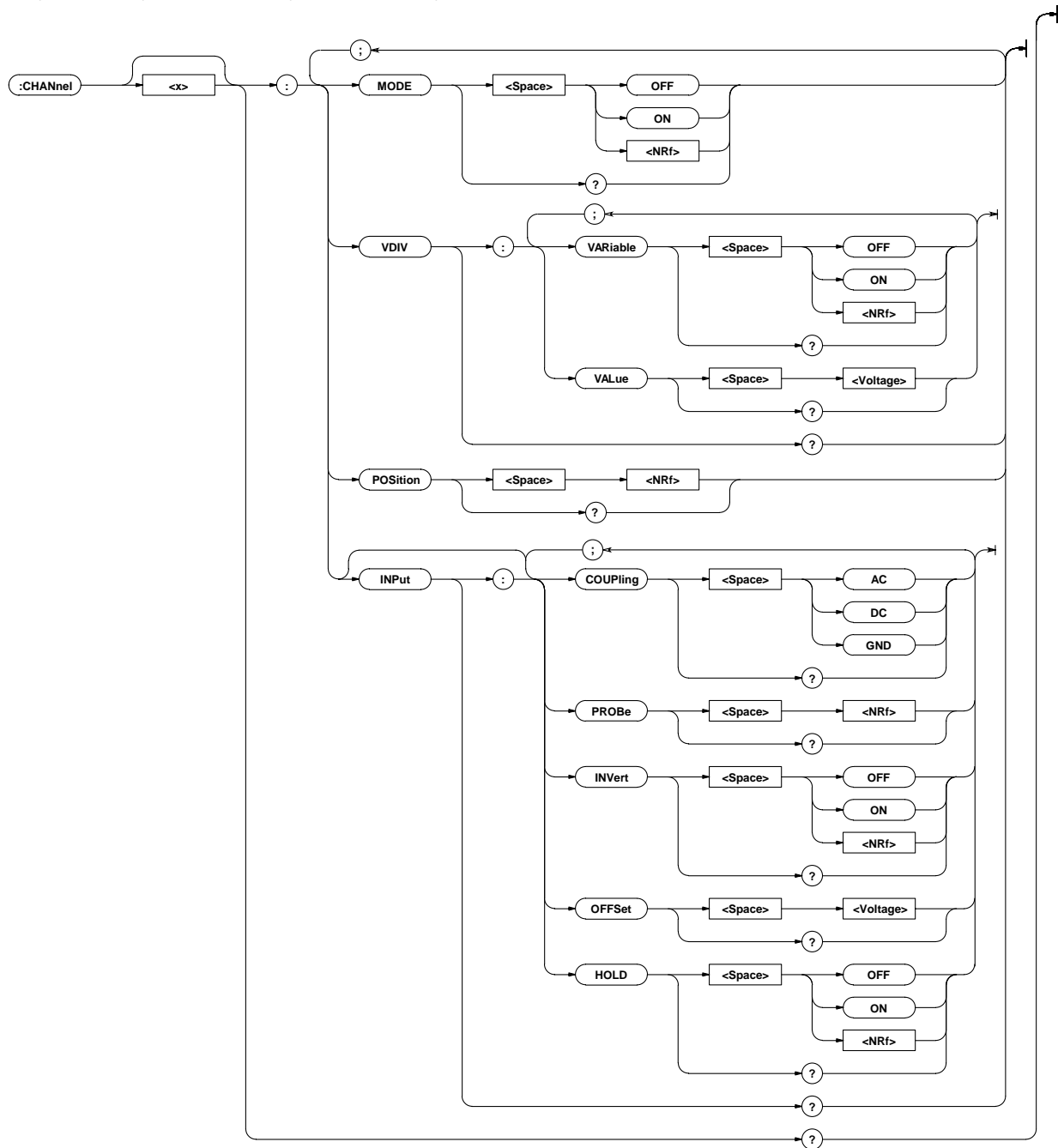
**Function** Executes calibration.  
**Syntax** CALibrate:EXECute  
**Example** CALIBRATE:EXECUTE

#### CALibrate:MODE

**Function** Turns auto calibration AUTO/OFF, or queries the current setting.  
**Syntax** CALibrate:MODE {AUTO|OFF}  
 CALibrate?  
**Example** CALIBRATE:MODE AUTO  
 CALIBRATE?→:CALIBRATE:MODE AUTO

## 4.5 CHANnel Group

The commands in the CHANnel group are used to make settings relating to, and inquiries about the vertical axis of the specified channel. This allows you to make the same settings and inquiries as when using the VERTICAL keys (CH key, V/DIV key, POSITION key and INPUT key).



### CHANnel<x>?

**Function** Queries all the vertical axis settings for the specified channel.

**Syntax** CHANnel<x>?  
<x>=1 to 4

**Example** CHANNEL1?→:CHANNEL1:MODE 1;INPUT:  
COUPLING DC;PROBE 10;INVERT 0;  
OFFSET 0.0E+00;HOLD 0;:CHANNEL1:  
POSITION 3.00;VDIV:VARIABLE OFF;  
VALUE 50.0E+00

### CHANnel<x>:INPut?

**Function** Queries all the vertical axis input conditions for the specified channel.

**Syntax** CHANnel<x>:INPut?  
<x>=1 to 4

**Example** CHANNEL1:INPUT?→:CHANNEL1:INPUT:  
COUPLING AC;PROBE 10;INVERT 0;  
OFFSET 0.0E+00;HOLD 0

**CHANnel<x>[:INPut]:COUPling**

**Function** Sets input coupling for the specified channel, or queries the current setting.

**Syntax** CHANnel<x>[:INPut]:COUPling {AC|DC|GND}  
CHANnel<x>[:INPut]:COUPling?  
<x>=1 to 4

**Example** CHANNEL1:INPUT:COUPLING AC  
CHANNEL1:INPUT:COUPLING?→:CHANNEL1:  
INPUT:COUPLING AC

**Description** If “DC” is selected, it is possible to set an offset voltage using “CHANnel<x>[:INPut]:OFFSet”.

**CHANnel<x>[:INPut]:HOLD**

**Function** Turns acquisition hold ON/OFF for the specified channel, or queries the current setting.

**Syntax** CHANnel<x>[:INPut]:HOLD {<Boolean>}  
CHANnel<x>[:INPut]:HOLD?  
<x>=1 to 4

**Example** CHANNEL1:INPUT:HOLD ON  
CHANNEL1:INPUT:HOLD?→:CHANNEL1:INPUT:  
HOLD 1

**CHANnel<x>[:INPut]:INVert**

**Function** Turns inversion of the waveform ON/OFF for the specified channel, or queries the current setting.

**Syntax** CHANnel<x>[:INPut]:INVert {<Boolean>}  
CHANnel<x>[:INPut]:INVert?  
<x>=1 to 4

**Example** CHANNEL1:INPUT:INVERT OFF  
CHANNEL1:INPUT:INVERT?→:CHANNEL1:INPUT:  
INVERT 0

**CHANnel<x>[:INPut]:OFFSet**

**Function** Sets the offset voltage for the specified channel, or queries the current setting.

**Syntax** CHANnel<x>[:INPut]:OFFSet <voltage>  
CHANnel<x>[:INPut]:OFFSet?  
<x>=1 to 4

<voltage>=for 10mV to 500mV/div  
-10V to 10V(1mV steps)  
for 1V to 5V/div  
-100V to 100V(10mV steps)  
for 10V to 50V/div  
-1kV to 1kV(100mV steps)

(when probe attenuation is 10:1)

**Example** CHANNEL1:INPUT:OFFSET 0V  
CHANNEL1:INPUT:OFFSET?→:CHANNEL1:INPUT:  
OFFSET 0.0E+00

**CHANnel<x>[:INPut]:PROBe**

**Function** Sets the probe attenuation for the specified channel, or queries the current setting.

**Syntax** CHANnel<x>[:INPut]:PROBe {<NRf>}  
CHANnel<x>[:INPut]:PROBe?  
<x>=1 to 4

{<NRf>}=1, 10, 100, 1000

**Example** CHANNEL1:INPUT:PROBE 10  
CHANNEL1:INPUT:PROBE?→:CHANNEL1:INPUT:  
PROBE 10

**CHANnel<x>:MODE**

**Function** Turns display ON/OFF for the specified channel, or queries the current setting.

**Syntax** CHANnel<x>:MODE {<Boolean>}  
CHANnel<x>:MODE?  
<x>=1 to 4

**Example** CHANNEL1:MODE ON  
CHANNEL1:MODE?→:CHANNEL1:MODE 1

**CHANnel<x>:POStion**

**Function** Sets the vertical position for the specified channel, or queries the current setting.

**Syntax** CHANnel<x>:POStion {<NRf>}  
CHANnel<x>:POStion?  
<x>=1 to 4  
{<NRf>}=-4.00 to 4.00 (div, 0.02 steps)

**Example** CHANNEL1:POSITION 0  
CHANNEL1:POSITION?→:CHANNEL1:  
POSITION 0.00

**CHANnel<x>:VDIV?**

**Function** Queries all the voltage axis sensitivity settings for the specified channel.

**Syntax** CHANnel<x>:VDIV?  
<x>=1 to 4

**Example** CHANNEL1:VDIV?→:CHANNEL1:VDIV:  
VARIABLE 0;VALUE 50.0E+00

**CHANnel<x>:VDIV:VALue**

**Function** Sets the voltage axis sensitivity for the specified channel, or queries the current setting.

**Syntax** CHANnel<x>:VDIV:VALue <voltage>  
CHANnel<x>:VDIV:VALue?  
<x>=1 to 4

<voltage>=10mV to 50V

**Example** CHANNEL1:VDIV:VALUE 50V  
CHANNEL1:VDIV:VALUE?→:CHANNEL1:VDIV:  
VALUE 50.0E+00

**CHANnel<x>:VDIV:VARiable**

**Function** Turns variable (fine adjustment) ON/OFF for the specified channel, or queries the current setting.

**Syntax** CHANnel<x>:VDIV:VARiable {<Boolean>}  
CHANnel<x>:VDIV:VARiable?  
<x>=1 to 4

**Example** CHANNEL1:VDIV:VARIABLE OFF  
CHANNEL1:VDIV:VARIABLE?→:CHANNEL1:VDIV:  
VARIABLE 0

## 4.6 CLEar Group

The CLEar command is used to clear the trace. This allows you to perform the same operation which you can perform using the CLEAR TRACE key on the front panel.

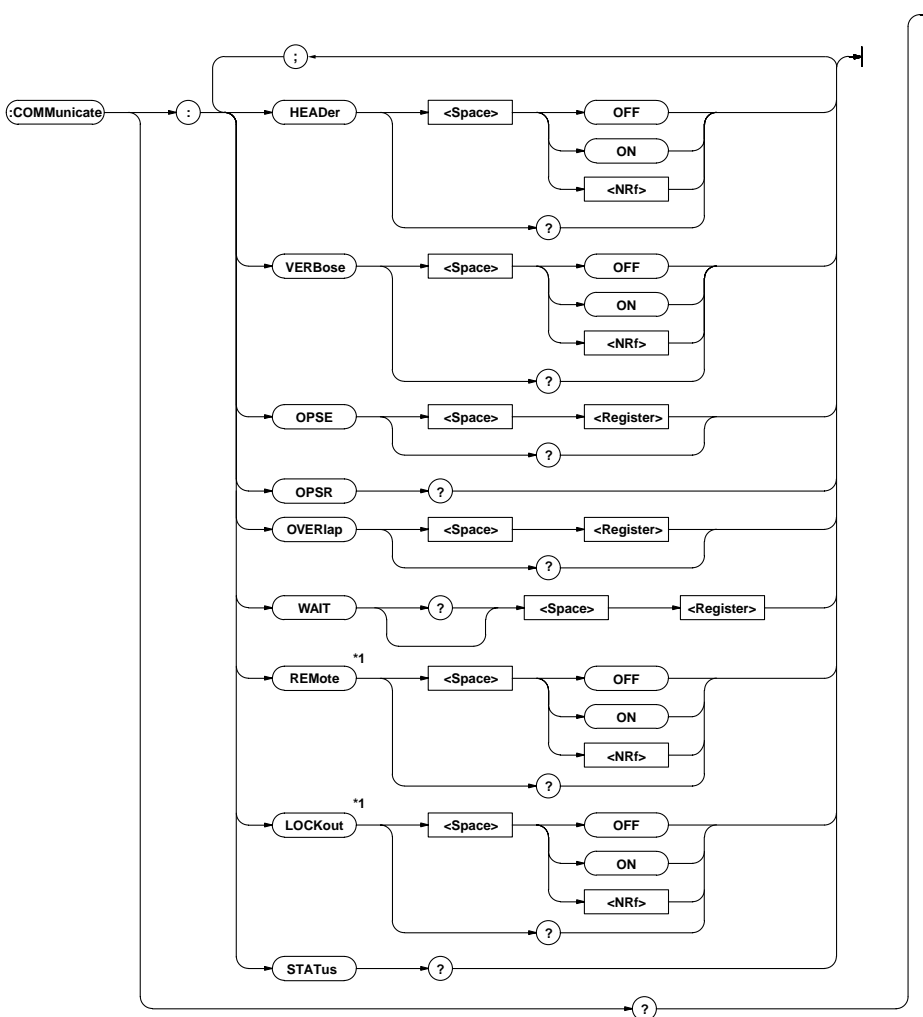


### CLEar

- Function** Clears the trace.
- Syntax** CLEar
- Example** CLEAR

## 4.7 COMMunicate Group

The commands in the COMMunicate group are used to make settings relating to, and inquires about communications. There is no front panel key for this function.



\*1 Available to use the separately sold RS-232-C interface unit 700927.

### COMMunicate?

- Function** Queries all the communication settings.
- Syntax** COMMunicate?
- Example** COMMUNICATE?→:COMMUNICATE:HEADER 1;  
OPSE65535;OVERLAP 65535;VERBOSE 1

### COMMunicate:HEADer

- Function** Determines whether a header is to be added (for example:"CHANNEL1:INPUT:PROBE 10") or not (for example:10) when sending a response to a query, or queries the current setting.
- Syntax** COMMunicate:HEADer {<Boolean>}  
COMMunicate:HEADer?
- Example** COMMUNICATE:HEADER ON  
COMMUNICATE:HEADER?→:COMMUNICATE:HEADER 1



**COMMunicate:LOCKout**

**Function** Sets local lockout ON or OFF.  
**Syntax** COMMunicate:LOCKout {<Boolean>}  
 COMMunicate:LOCKout?  
**Example** COMMUNICATE:LOCKOUT ON  
 COMMUNICATE:LOCKOUT?→:COMMUNICATE:  
 LOCKOUT 1  
**Description** This command is available to use the separately sold RS-232-C interface unit 700927.

**COMMunicate:OPSE**

**Function** Sets the overlap command to be used with \*OPC, \*OPC?, and \*WAI, or queries the current setting.  
**Syntax** COMMunicate:OPSE <Register>  
 COMMunicate:OPSE?  
 <Register>=0 to 65535, refer to figure 4.1  
**Example** COMMUNICATE:OPSE #HFFFF  
 COMMUNICATE:OPSE?→:COMMUNICATE:OPSE 240

**COMMunicate:OPSR?**

**Function** Returns the value in the operation pending status register.  
**Syntax** COMMunicate:OPSR?  
**Example** COMMUNICATE:OPSR?→0  
**Description** For a description of the operation pending status register, refer to figure 4.1.

**COMMunicate:OVERlap**

**Function** Selects the command to be used for overlap, or queries the current setting.  
**Syntax** COMMunicate:OVERlap <Register>  
 COMMunicate:OVERlap?  
 <Register>=0 to 65535, refer to figure 4.1  
**Example** COMMUNICATE:OVERLAP #HFFFF  
 COMMUNICATE:OVERLAP?→:COMMUNICATE:  
 OVERLAP 176  
**Description** For a description of the synchronization method using "COMMunicate:OVERlap", refer to page 3-7.

**COMMunicate:REMOte**

**Function** Sets remote (ON) or local mode (OFF).  
**Syntax** COMMunicate:REMOte {<Boolean>}  
 COMMunicate:REMOte?  
**Example** COMMUNICATE:REMOTE ON  
 COMMUNICATE:REMOTE?→:COMMUNICATE:  
 REMOTE 1  
**Description** This command is available to use the separately sold RS-232-C interface unit 700927.

**COMMunicate:STATus?**

**Function** Queries the status of a specified circuit.  
**Syntax** COMMunicate:STATus?  
**Example** COMMUNICATE:STATUS?→:COMMUNICATE:  
 STATUS 0

**Description** The status condition for each bit is as follows.

bit	GP-IB	RS-232-C
0	permanent comm. error	Parity error
1	always 0	framing error
2	always 0	break character occurrence
3 and up	always 0	always 0

When a status occurs which results in changing of the bits, reading it will clear the error.

**COMMunicate:VERBose**

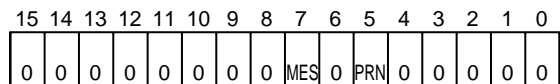
**Function** Determines whether a response to a query is to be returned in full form (for example: CHANNEL1:INPUT:PROBE 10), or in abbreviated form (for example: CHAN1:PROB 10), or queries the current setting.  
**Syntax** COMMunicate:VERBose {<Boolean>}  
 COMMunicate:VERBose?  
**Example** COMMUNICATE:VERBOSE ON  
 COMMUNICATE:VERBOSE?→:COMMUNICATE:  
 VERBOSE 1

**COMMunicate:WAIT**

**Function** Waits until one of the specified extended event occurs.  
**Syntax** COMMunicate:WAIT <Register>  
 <Register>=0 to 65535 (For a description of the extended event register, refer to page 5-4.)  
**Example** COMMUNICATE:WAIT 65535  
**Description** For a description of synchronization using "COMMunicate:WAIT", refer to page 3-8.

**COMMunicate:WAIT?**

**Function** Generates a response when one of the specified extended events occurs.  
**Syntax** COMMunicate:WAIT? <Register>  
 <Register>=0 to 65535 (For a description of the extended event register, refer to page 5-4.)  
**Example** COMMUNICATE:WAIT? 65535→1

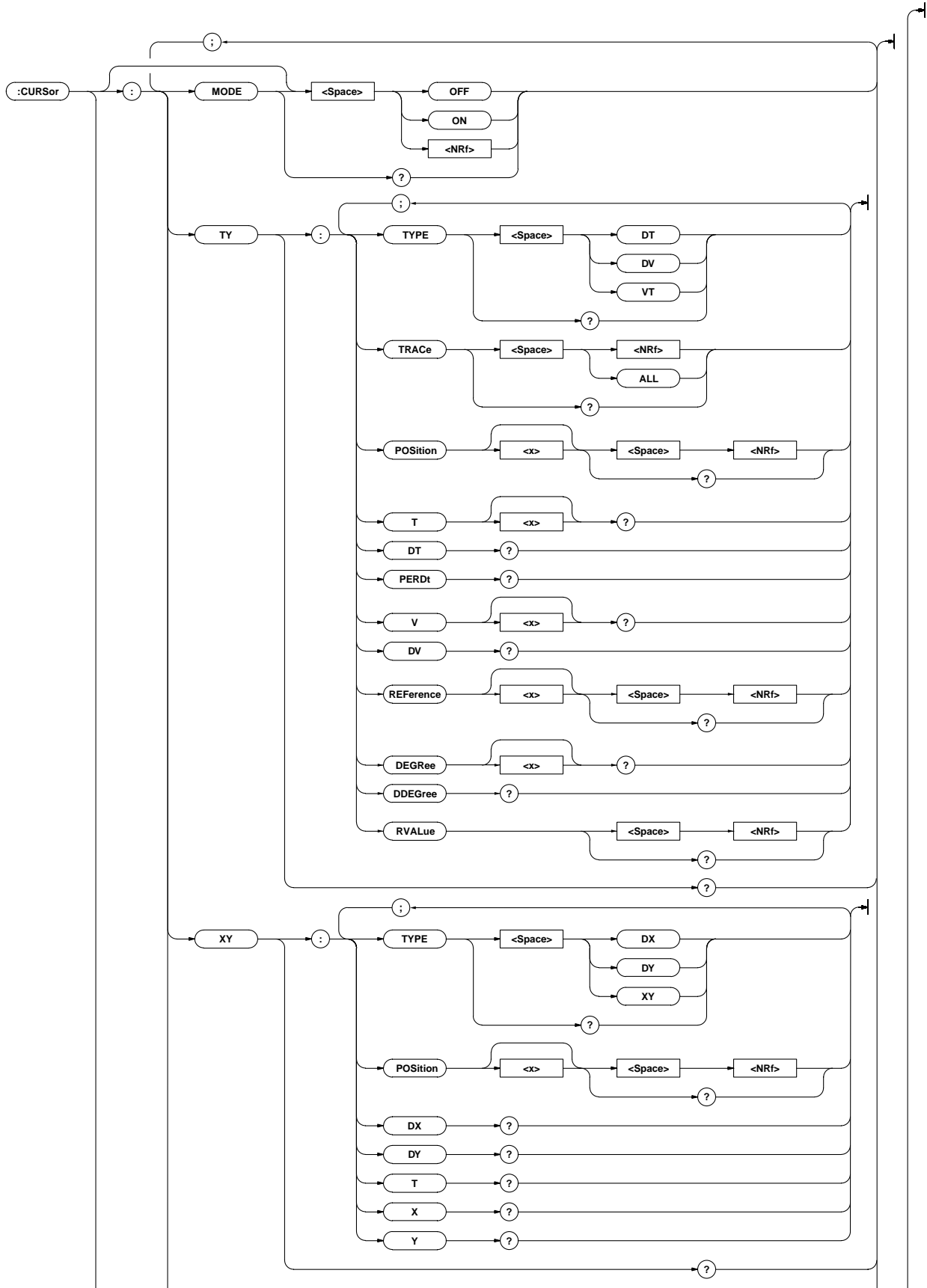


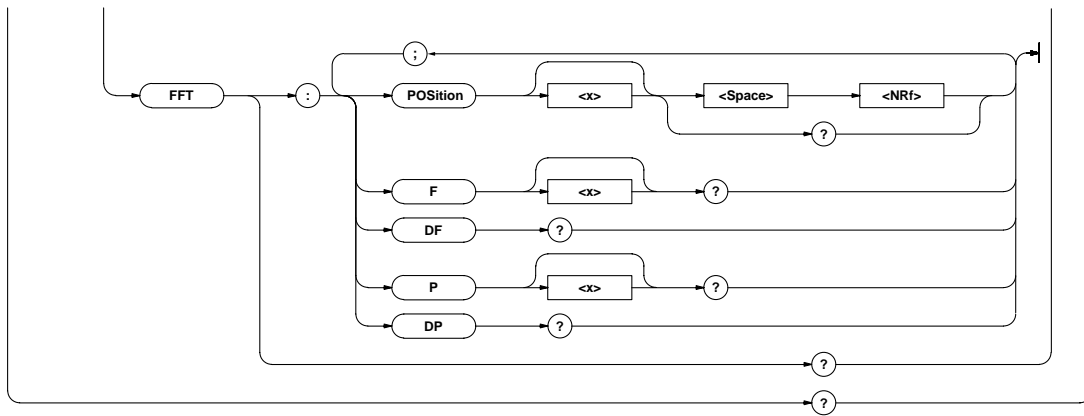
when bit 5(PRN)=1 : printer operation is not completed  
 when bit 7(MES)=1 : Auto-measurement of ACQ data is not completed

**fig 4.1** Operation pending status register/overlap enable register.

### 4.8 CURSOR Group

The commands in the CURSOR group are used to make settings relating to, and inquiries about cursor measurement. This allows you to make the same settings and inquiries as when using the CURSOR key on the front panel.





**CURSor?**

**Function** Queries all the settings relating to the FFT cursors.

**Syntax** CURSor?

**Example** CURSOR?→:CURSOR:MODE 0;TY:TYPE DT;  
TRACE 1;REFERENCE1 -1.50;REFERENCE2 1.50;  
POSITION1 -2.50;POSITION2 2.50;  
RVALUE 360;:CURSOR:XY:TYPE DX;  
POSITION1 -2.50;POSITION2 2.50;:CURSOR:  
FFT:POSITION1 -2.50;POSITION2 2.50

**CURSor:FFT?**

**Function** Queries all the FFT cursor measurement settings.

**Syntax** CURSor:FFT?

**Example** CURSOR:FFT?→:CURSOR:FFT:POSITION1 -2.50;  
POSITION2 2.50

**CURSor:FFT:DF?**

**Function** Queries the DF value between the FFT cursors.

**Syntax** CURSor:FFT:DF?

**Example** CURSOR:FFT:DF?→:CURSOR:FFT:DF 200.00E+03

**CURSor:FFT:DP?**

**Function** Queries the DP value between the FFT cursors.

**Syntax** CURSor:FFT:DP?

**Example** CURSOR:FFT:DP?→:CURSOR:FFT:DP 20.00E+00

**CURSor:FFT:F<x>?**

**Function** Queries the frequency values of the FFT cursors.

**Syntax** CURSor:FFT:F<x>?  
<x>=1,2

**Example** (example is shown for the F1 cursor)  
CURSOR:FFT:F1?→:CURSOR:FFT:F1 100.00E+03

**CURSor:FFT:P<x>?**

**Function** Queries the P values of the FFT cursors.

**Syntax** CURSor:FFT:P<x>?  
<x>=1,2

**Example** (example is shown for the F1 cursor)  
CURSOR:FFT:P1?→:CURSOR:FFT:P1 20.0E+00

**CURSor:FFT:POsition<x>**

**Function** Sets the position of the FFT cursor, or queries the current setting.

**Syntax** CURSor:FFT:POsition<x> {<NRf>}  
CURSor:FFT:POsition<x>?  
<x>=1,2  
{<NRf>}=-5.00 to 5.00(div, 0.02 steps)

**Example** (example is shown for the F1 cursor)  
CURSOR:FFT:POSITION1 2.00  
CURSOR:FFT:POSITION1?→:CURSOR:FFT:  
POSITION1 2.00

**CURSor[:MODE]**

**Function** Sets cursor measurement ON/OFF, or queries the current settings.

**Syntax** CURSor[:MODE] {<Boolean>}  
CURSor:MODE?

**Example** CURSOR:MODE ON  
CURSOR:MODE?→:CURSOR:MODE 1

**CURSor:TY?**

**Function** Queries all the T-Y cursor measurement settings.

**Syntax** CURSor:TY?

**Example** CURSOR:TY?→:CURSOR:TY:TYPE VT;TRACE 1;  
REFERENCE1 -1.50;REFERENCE2 1.50;  
POSITION1 -2.50;POSITION2 2.50;RVALUE 360

**CURSor:TY:DDEGREE?**

**Function** Queries the phase difference value, ΔD, between D1 and D2 phase cursors.

**Syntax** CURSor:TY:DDEGREE?

**Example** CURSOR:TY:DDEGREE?→:CURSOR:TY:  
DDEGREE 600.000E+00

**CURSor:TY:DEGREE<x>?**

**Function** Queries the value of D1 or D2 cursor.

**Syntax** CURSor:TY:DEGREE<x>?  
<x>=1, 2

**Example** (example is shown for the D1 cursor)  
CURSOR:TY:DEGREE1?→:CURSOR:TY:  
DEGREE1 -120.000E+00

**CURSor:TY:DT?**

**Function** Queries the DT value between the T-Y cursors.

**Syntax** CURSor:TY:DT?

**Example** CURSOR:TY:DT?→:CURSOR:TY:DT 200.00E-03

**CURSOR:TY:DV?**

**Function** Queries the DV value between the T-Y cursors.

**Syntax** CURSOR:TY:DV?

**Example** CURSOR:TY:DV?→:CURSOR:TY:DV 50.0E+00

**CURSOR:TY:PERDt?**

**Function** Queries the 1/DT value between the T-Y cursors.

**Syntax** CURSOR:TY:PERDt?

**Example** CURSOR:TY:PERDt?→:CURSOR:TY:  
PERDt 200.00E+03

**CURSOR:TY:POsition<x>**

**Function** Sets the position of the T-Y cursor, or queries the current setting.

**Syntax** CURSOR:TY:POsition<x> {<NRF>}

CURSOR:TY:POsition<x>?

<x>=1,2

{<NRF>}=-5.00 to 5.00(div,0.01 step;when  
cursor type is DT, VT or DEGRee)  
=-4.00 to 4.00(div,0.02 step;when  
cursor type isDV)  
=-5.0000 to 5.0000 (div: when  
cursor type of DL1540C is VTACq)  
=-5.00000 to 5.00000 (div: when  
cursor type of DL1540CL is VTACq)

**Example** (example is shown for the T1 cursor)

CURSOR:TY:POSITION1 2.00

CURSOR:TY:POSITION1?→:CURSOR:TY:  
POSITION1 2.00

**CURSOR:TY:REfERENCE<x>**

**Function** Sets the position of the start point or end point (R1, R2) for the phase reference, or queries the current setting.

**Syntax** CURSOR:TY:REfERENCE<x> {<NRF>}

CURSOR:TY:REfERENCE<x>?

<x>=1,2

{<NRF>}=-15.00 to 5.00 (div, 0.01 step)

**Example** (example is shown for the R1 cursor)

CURSOR:TY:REFERENCE1→2.00

CURSOR:TY:REFERENCE1?→:CURSOR:TY:  
REFERENCE1→2.00

**CURSOR:TY:RVALue**

**Function** Sets the phase reference value between the vertical cursors (phase cursors), or queries the current setting.

**Syntax** CURSOR:TY:RVALue {<NRF>}

CURSOR:TY:RVALue?

{<NRF>}=180,360,540,720

**Example** CURSOR:TY:RVALUE 180

CURSOR:TY:RVALUE?→:CURSOR:TY:RVALUE 180

**CURSOR:TY:TRACe**

**Function** Sets the waveform of measurement of the T-Y cursor, or queries the current setting.

**Syntax** CURSOR:TY:TRACe {{<NRF>}|ALL}

CURSOR:TY:TRACe?

{<NRF>}=1 to 8 (5 to 8 corresponds to  
Load1 to Load4)

**Example** CURSOR:TY:TRACE 1

CURSOR:TY:TRACE?→:CURSOR:TY:TRACE 1

**CURSOR:TY:TYPE**

**Function** Sets the measurement type of the T-Y cursor, or queries the current setting.

**Syntax** CURSOR:TY:TYPE {DT|DV|VT|VTACq|DEGRee}

CURSOR:TY:TYPE?

**Example** CURSOR:TY:TYPE VT

CURSOR:TY:TYPE?→:CURSOR:TY:TYPE VT

**CURSOR:TY:T<x>?**

**Function** Queries the time corresponding to the T-Y cursor.

**Syntax** CURSOR:TY:T<x>?

<x>=1,2

**Example** (example is shown for the T1 cursor)

CURSOR:TY:T1?→:CURSOR:TY:T1 100.00E-03

**CURSOR:TY:V<x>?**

**Function** Queries the voltage value corresponding to the T-Y cursor.

**Syntax** CURSOR:TY:V<x>?

<x>=1,2

**Example** (example is shown for the T1 and V1 cursor)

CURSOR:TY:V1?→:CURSOR:TY:V1 50.0E+00

However, when ALL traces are selected, a V1 query results in a response similar to the following.

CURSOR:TY:V1?→:CURSOR:TY:V1 50.0E+00,  
50.0E+00,50.0E+00,50.0E+00,50.0E+00,  
50.0E+00,50.0E+00,50.0E+00

**CURSOR:XY?**

**Function** Queries all the X-Y cursor measurement settings.

**Syntax** CURSOR:XY?

**Example** CURSOR:XY?→:CURSOR:XY:TYPE XY;  
POSITION1 -2.50;POSITION2 2.50

**CURSOR:XY:DX?**

**Function** Queries the DX value between the X-Y cursors.

**Syntax** CURSOR:XY:DX?

**Example** CURSOR:XY:DX?→:CURSOR:XY:DX 50.0E+00

**CURSOR:XY:DY?**

**Function** Queries the DY value between the X-Y cursors.

**Syntax** CURSOR:XY:DY?

**Example** CURSOR:XY:DY?→:CURSOR:XY:DY 50.0E+00

**CURSOR:XY:POsition<x>**

**Function** Sets the position of the X-Y cursor, or queries the current setting.

**Syntax** CURSOR:XY:POsition<x> {<NRF>}

CURSOR:XY:POsition<x>?

<x>=1,2

{<NRF>}=-5.00 to 5.00(div, cursor type  
DX:0.02 step, cursor type XY:0.01  
step)

=-4.00 to 4.00(div, 0.02 step,  
when cursor type is DY)

**Example** (example is shown for the X1 cursor)

CURSOR:XY:POSITION1 2.00

CURSOR:XY:POSITION1?→:CURSOR:XY:  
POSITION1 2.00

## 4.8 CURSor Group

---

### **CURSor:XY:T?**

**Function** Queries the time corresponding to the X-Y cursor.

**Syntax** CURSor:XY:T?

**Example** CURSOR:XY:T?→:CURSOR:XY:T 100.00E-03

### **CURSor:XY:TYPE**

**Function** Sets the measurement type of the X-Y cursor, or queries the current setting.

**Syntax** CURSor:XY:TYPE {DX|DY|XY}  
CURSor:XY:TYPE?

**Example** CURSOR:XY:TYPE XY  
CURSOR:XY:TYPE?→:CURSOR:XY:TYPE XY

### **CURSor:XY:X?**

**Function** Queries the voltage value corresponding to the X waveform of the X-Y cursor.

**Syntax** CURSor:XY:X?

**Example** CURSOR:XY:X?→:CURSOR:XY:X 50.0E+00

### **CURSor:XY:Y?**

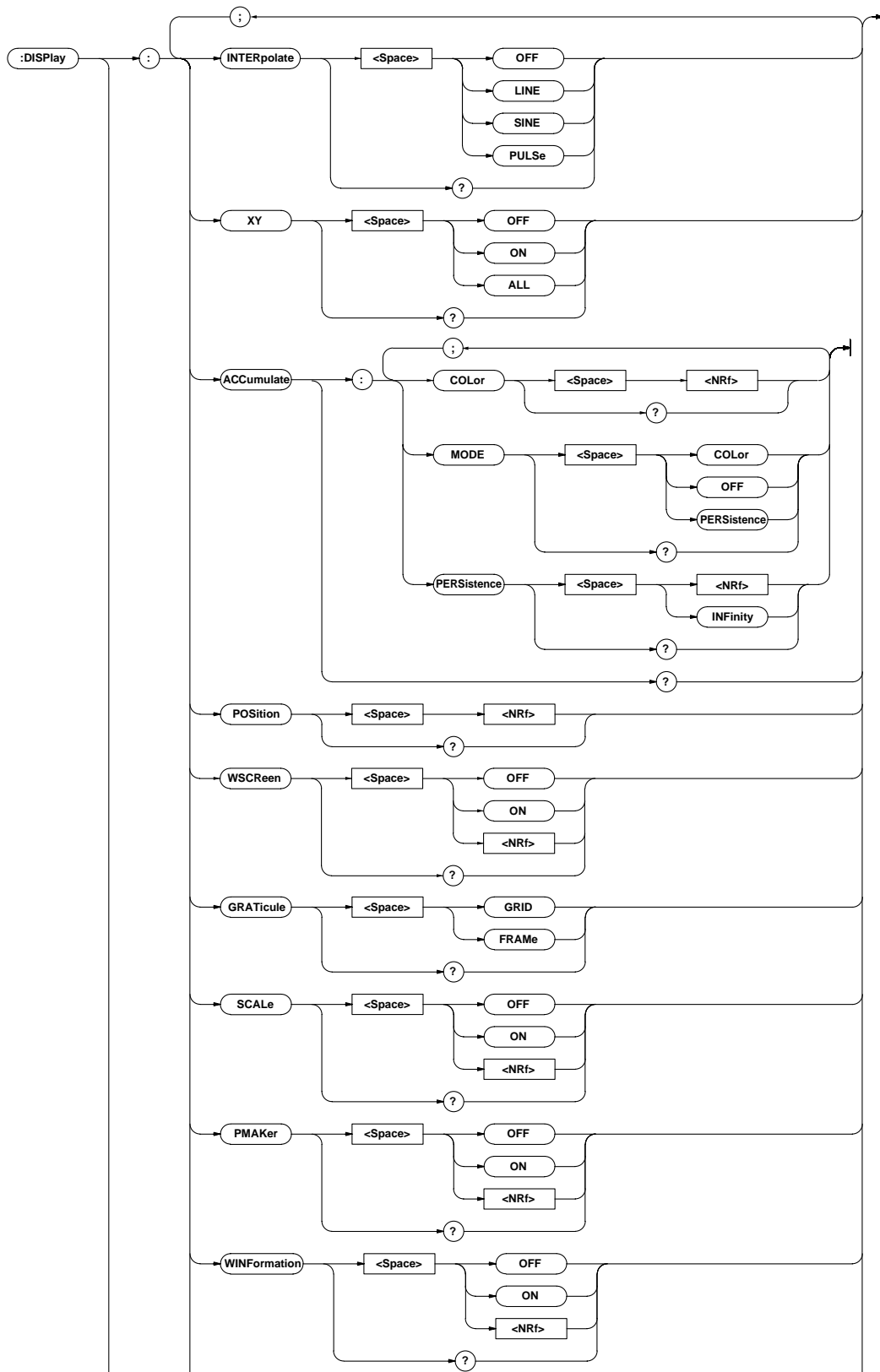
**Function** Queries the voltage value corresponding to the Y waveform of the X-Y cursor.

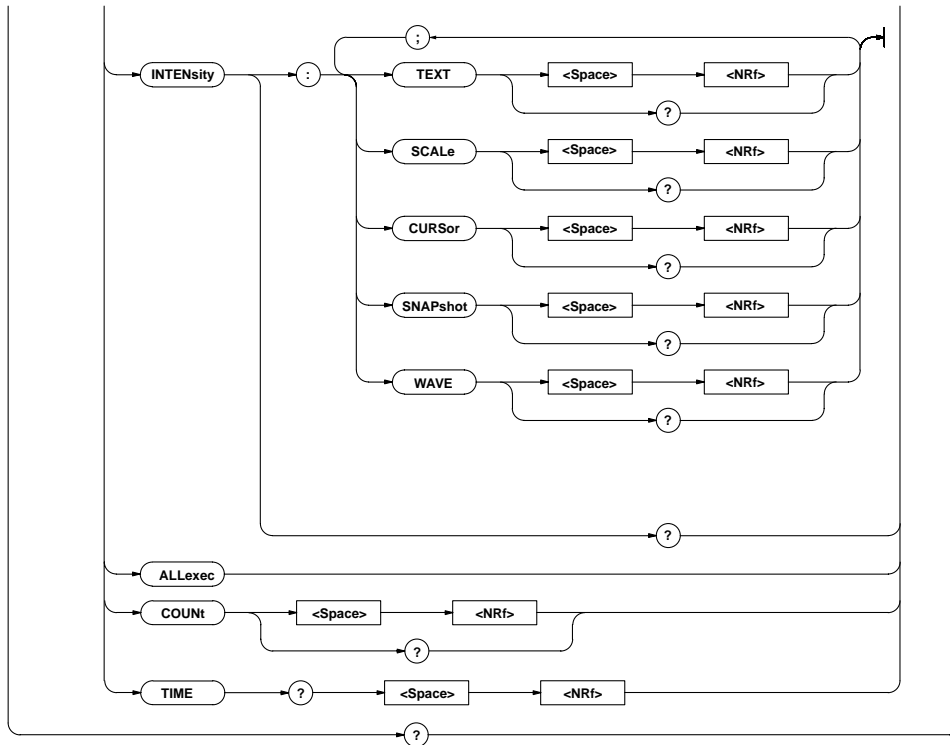
**Syntax** CURSor:XY:Y?

**Example** CURSOR:XY:Y?→:CURSOR:XY:Y 50.0E+00

## 4.9 DISPlay Group

The commands in the DISPlay group are used to make settings relating to, and inquiries about display. This allows you to make the same settings and inquiries as when using the DISPLAY, POSITION, and such keys.





**DISPlay?**

**Function** Queries all the display settings.  
**Syntax** DISPlay?  
**Example** DISPlay?→:DISPlay:INTERPOLATE SINE;  
 XY OFF;ACCUMULATE:MODE OFF;COLOR 8;  
 PERSISTANCE 100.0E-03;DISPlay:  
 TRACE EACH;COUNT 0;POSITION 0.0000;  
 WSCREEN 1;GRATICULE GRID;SCALE 1;  
 WINFORMATION 1;INTENSITY:TEXT 9;SCALE 2;  
 CURSOR 10;SNAPSHOT 5;WAVE 8;:DISPlay:  
 PMAKER 0

**DISPlay:ACCumulate?**

**Function** Queries all the accumulation settings.  
**Syntax** DISPlay:ACCumulate?  
**Example** DISPlay:ACCUMULATE?→:DISPlay:ACCUMULATE:  
 MODE OFF;COLOR 8;PERSISTANCE 100.0E-03

**DISPlay:ACCumulate:COLor**

**Function** Sets/queries the color grading width.  
**Syntax** DISPlay:ACCumulate:COLor {<NRf>}  
 DISPlay:ACCumulate:COLor?  
 <NRf>≥1 to 2048 (in steps of 2<sup>n</sup>)  
**Example** DISPlay:ACCUMULATE:COLOR 16  
 DISPlay:ACCUMULATE:COLOR?→:DISPlay:  
 ACCUMULATE:COLOR 16

**DISPlay:ACCumulate:MODE**

**Function** Turns accumulation mode ON/OFF, or queries the current setting.  
**Syntax** DISPlay:ACCumulate:MODE  
 {OFF|PERSISTANCE|COLOR}  
 DISPlay:ACCumulate:MODE?  
**Example** DISPlay:ACCUMULATE:MODE OFF  
 DISPlay:ACCUMULATE:MODE?→:DISPlay:  
 ACCUMULATE:MODE OFF

**DISPlay:ACCumulate:PERSistance**

**Function** Sets the accumulation time, or queries the current setting.  
**Syntax** DISPlay:ACCumulate:PERSistance  
 {<time>|INFINITY}  
 DISPlay:ACCumulate:PERSistance?  
 <time>= 100ms to 900ms(100ms step)  
 1s to 9s(1s step)  
 10s to 60s(10s step)  
**Example** DISPlay:ACCUMULATE:PERSISTANCE 0.15  
 DISPlay:ACCUMULATE:PERSISTANCE?→:  
 DISPlay:ACCUMULATE:PERSISTANCE 100.0E-03

**DISPlay:ALLExec**

**Function** Displays all stored waveforms.  
**Syntax** DISPlay:ALLExec  
**Example** DISPlay:ALLEXEC  
**Description** To be used when the trigger mode is N-SGL, or when the history memory function is ON; cannot be used while acquisition is in progress.

**DISPlay:COUnT**

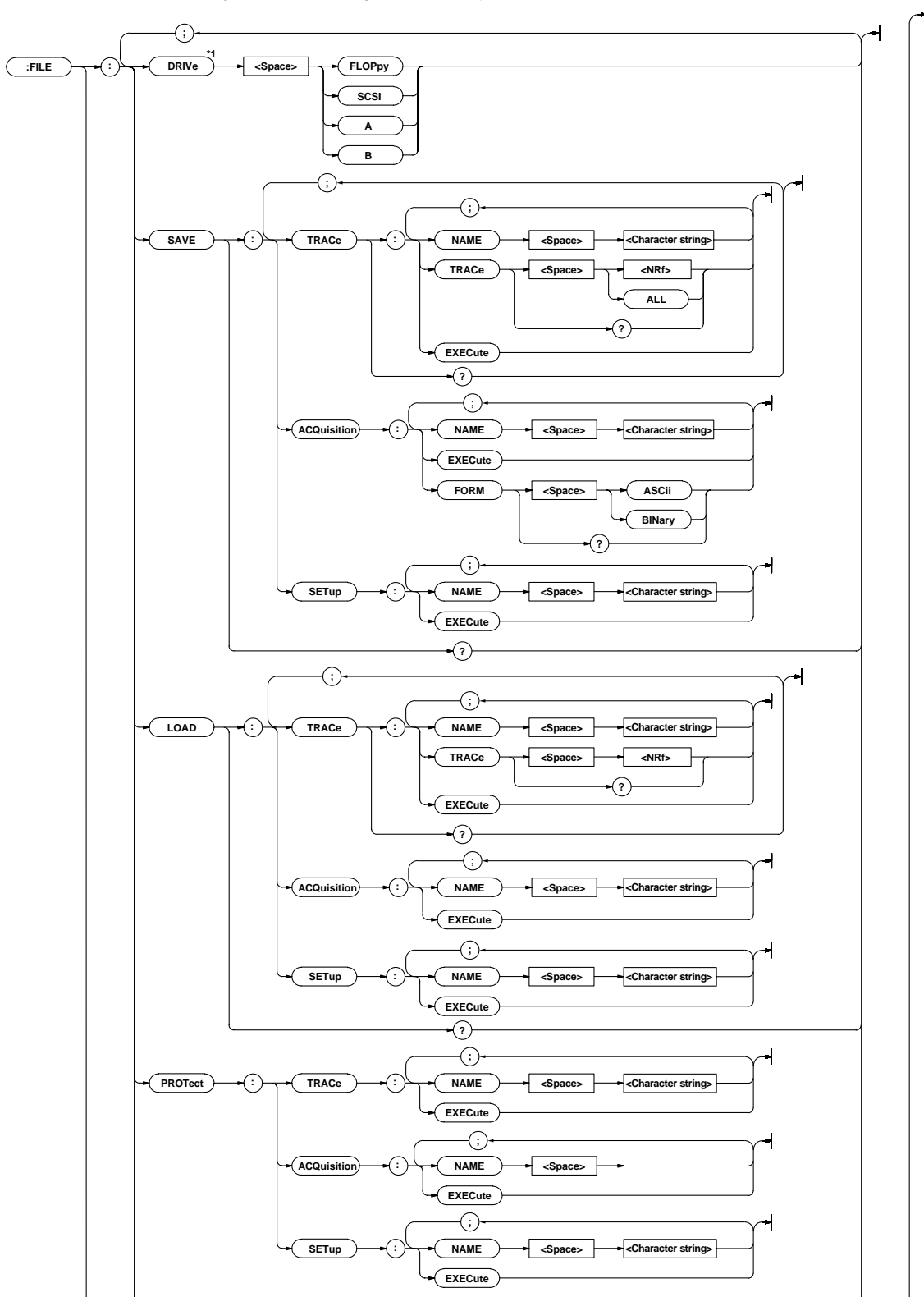
**Function** Sets the number of the waveform to be displayed, or queries the current setting.  
**Syntax** DISPlay:COUnT {<NRf>}  
 DISPlay:COUnT?  
 {<NRf>}=0 to -99  
**Example** DISPlay:COUNT -1  
 DISPlay:COUNT?→:DISPlay:COUNT -1  
**Description** To be used when the trigger mode is N-SGL, or when the history memory function is ON; cannot be used while acquisition is in progress.

**DISPlay:GRATicule****Function** Sets the graticule, or queries the current setting.**Syntax** DISPlay:GRATicule {FRAME|GRID}  
DISPlay:GRATicule?**Example** DISPLAY:GRATICULE GRID  
DISPLAY:GRATICULE?→:DISPLAY:  
GRATICULE GRID**DISPlay:INTENsity?****Function** Queries the intensity of displayed parameters.**Syntax** DISPlay:INTENsity?**Example** DISPLAY:INTENSITY?→:DISPLAY:INTENSITY:  
ALL 8;TEXT 9;SCALE 6;CURSOR 10;  
SNAPSHOT 5;WAVE 8**DISPlay:INTENsity:{CURSor|SCALE|  
SNAPshot|TEXT|WAVE}****Function** Sets the intensity of each displayed parameter,  
or queries the current setting.**Syntax** DISPlay:INTENsity:{CURSor|SCALE|SNAPshot|  
TEXT|WAVE} {<Nrf>}  
DISPlay:INTENsity:{CURSor|SCALE|SNAPshot|  
TEXT|WAVE}?  
{<Nrf>}=1 to 15**Example** (example is shown for the cursor)  
DISPLAY:INTENSITY:CURSOR 6  
DISPLAY:INTENSITY:CURSOR?→:DISPLAY:  
INTENSITY:CURSOR 6**DISPlay:INTERPolate****Function** Sets the interpolation method, or queries the  
current setting.**Syntax** DISPlay:INTERPolate {LINE|OFF|PULSe|SINE}  
DISPlay:INTERPolate?**Example** DISPLAY:INTERPOLATE OFF  
DISPLAY:INTERPOLATE?→:DISPLAY:  
INTERPOLATE OFF**DISPlay:PMARker****Function** Sets the % marker ON/OFF, or queries the  
current setting.**Syntax** DISPlay:PMARker {<Boolean>}  
DISPlay:PMARker?**Example** DISPLAY:PMARKER ON  
DISPLAY:PMARKER?→:DISPLAY:PMARKER 1**DISPlay:POSition****Function** Sets the horizontal/vertical position of the  
waveform, or queries the current setting.**Syntax** DISPlay:POSition {<Nrf>}  
DISPlay:POSition?  
{<Nrf>}=-5.0000 to 5.0000(div, 0.0002  
step)**Example** DISPLAY:POSITION 1.0000  
DISPLAY:POSITION?→:DISPLAY:  
POSITION 1.0000**DISPlay:SCALe****Function** Turns display of the scale ON/OFF, or queries  
the current setting.**Syntax** DISPlay:SCALe {<Boolean>}  
DISPlay:SCALe?**Example** DISPLAY:SCALE ON  
DISPLAY:SCALE?→:DISPLAY:SCALE 1**DISPlay:TIME?****Function** Queries the time of acquisition corresponding to  
the waveform with the display No..**Syntax** DISPlay:TIME? {<Nrf>}  
{<Nrf>}=0 to -99**Example** DISPLAY:TIME? -1→:DISPLAY:  
TIME "#-01 10:21:18"**Description** To be used when the trigger mode is N-SGL, or  
when the history memory function is ON; cannot  
be used while acquisition is in progress.**DISPlay:WINformation****Function** Turns display of waveform information ON/OFF,  
or queries the current setting.**Syntax** DISPlay:WINformation {<Boolean>}  
DISPlay:WINformation?**Example** DISPLAY:WINFORMATION ON  
DISPLAY:WINFORMATION?→:DISPLAY:  
WINFORMATION 1**DISPlay:WSCREEN****Function** Turns the wide screen mode ON/OFF, or queries  
the current setting.**Syntax** DISPlay:WSCREEN {<Boolean>}  
DISPlay:WSCREEN?**Example** DISPLAY:WSCREEN ON  
DISPLAY:WSCREEN?→:DISPLAY:WSCREEN 1**DISPlay:XY****Function** Turns display of the X-Y waveform ON/OFF, or  
queries the current setting.**Syntax** DISPlay:XY {OFF|ON|ALL}  
DISPlay:XY?**Example** DISPLAY:XY OFF  
DISPLAY:XY?→:DISPLAY:XY 0

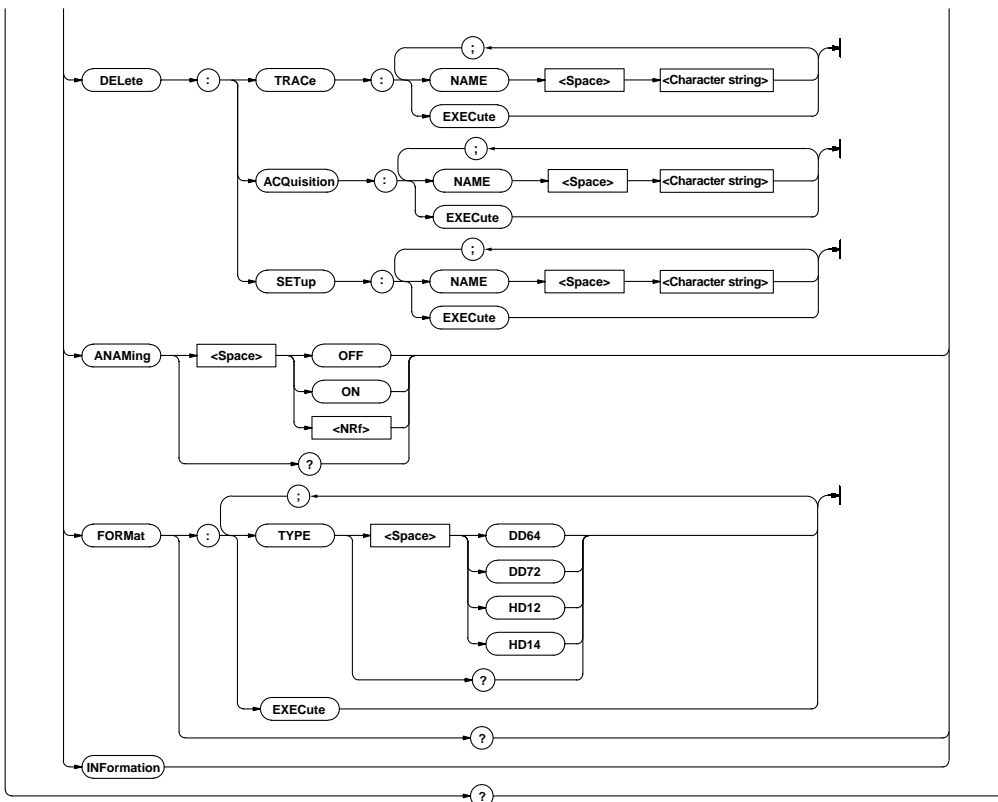


### 4.10 FILE Group

The commands in the FILE group are used to make settings relating to, or inquires about a floppy disk. This allows you to make the same settings as when using the FILE key on the front panel.



\*1 Available to use the separately sold SCSI interface unit 700930.

**FILE?**

**Function** Queries all the settings relating to files.

**Syntax** FILE?

**Example** FILE?→:FILE:SAVE:TRACE:TRACE 1;;FILE:SAVE:ACQUISITION:TRACE 1;;FILE:LOAD:TRACE:TRACE 1;;FILE:FORMAT:TYPE HD12

**FILE:ANAMing**

**Function** Sets the auto naming mode for file names, or queries the current setting.

**Syntax** FILE:ANAMing {<Boolean>}  
FILE:ANAMing?

**Example** FILE:ANAMING ON  
FILE:ANAMING?→:FILE:ANAMING 1

**FILE:DELEte:ACQuisition:EXECute**

**Function** Executes the deletion of an acquisition data file.

**Syntax** FILE:DELEte:ACQuisition:EXECute

**Example** FILE:DELETE:ACQUISITION:EXECUTE

**FILE:DELEte:ACQuisition:NAME**

**Function** Sets the name of the acquisition data file to be deleted.

**Syntax** FILE:DELEte:ACQuisition:  
NAME<character string>  
<character string>= Up to 8 characters  
can be used, refer to User's manual  
IM701530-01E.

**Example** FILE:DELETE:ACQUISITION:NAME "ACQ\_1"

**FILE:DELEte:SETup:EXECute**

**Function** Executes the deletion of an setting parameter data file.

**Syntax** FILE:DELEte:SETup:EXECute

**Example** FILE:DELETE:SETUP:EXECUTE

**FILE:DELEte:SETup:NAME**

**Function** Sets the name of the setting parameter data file to be deleted.

**Syntax** FILE:DELEte:SETup:NAME <character string>  
<character string>= Up to 8 characters  
can be used, refer to User's manual  
IM701530-01E.

**Example** FILE:DELETE:SETUP:NAME "SETUP\_1"

**FILE:DELEte:TRACe:EXECute**

**Function** Executes the deletion of a displayed waveform file.

**Syntax** FILE:DELEte:TRACe:EXECute

**Example** FILE:DELETE:TRACE:EXECUTE

**FILE:DELEte:TRACe:NAME**

**Function** Sets the name of the displayed waveform file to be deleted.

**Syntax** FILE:DELEte:TRACe:NAME <character string>  
<character string>= Up to 8 characters  
can be used, refer to User's manual  
IM701530-01E.

**Example** FILE:DELETE:TRACE:NAME "TRACE\_1"

**FILE:DRIVE**

**Function** Sets the target media to control with the FILE Group Commands, or queries the current setting. FLOPPy or A sets the floppy disk and SCSI or B sets SCSI. The response to an inquiry is given in terms of A or B.

**Syntax** FILE:DRIVE {A|B|FLOPPy|SCSI}  
FILE:DRIVE?

**Example** FILE:DRIVE FLOPPY  
FILE:DRIVE?→:FILE:DRIVE A

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

**FILE:FORMat?**

**Function** Queries all the settings relating to floppy disk formatting.

**Syntax** FILE:FORMat?

**Example** FILE:FORMAT?→:FILE:FORMAT:TYPE HD12

**FILE:FORMat:EXECute**

**Function** Formats the floppy disk.

**Syntax** FILE:FORMat:EXECute

**Example** FILE:FORMAT:EXECUTE

**FILE:FORMat:TYPE**

**Function** Selects the floppy disk format to be used for formatting, or queries the current setting.

**Syntax** FILE:FORMat:TYPE {DD64|DD72|HD12|HD14}  
FILE:FORMat:TYPE?

**Example** FILE:FORMAT:TYPE HD12  
FILE:FORMAT:TYPE?→:FILE:FORMAT:TYPE HD12

**FILE:INFormation**

**Function** Checks floppy disk file information.

**Syntax** FILE:INFormation

**FILE:LOAD?**

**Function** Queries the settings relating to loading data.

**Syntax** FILE:LOAD?

**Example** FILE:LOAD?→:FILE:LOAD:TRACE:TRACE1

**FILE:LOAD:ACQuisition:EXECute**

**Function** Executes the loading of acquisition waveform data.

**Syntax** FILE:LOAD:ACQuisition:EXECute

**Example** FILE:LOAD:ACQUISITION:EXECUTE

**FILE:LOAD:ACQuisition:NAME**

**Function** Sets the name of the acquisition waveform data file to be loaded.

**Syntax** FILE:LOAD:ACQuisition:  
NAME <character string>  
<character string>= Up to 8 characters  
can be used, refer to User's manual  
IM701530-01E.

**Example** FILE:LOAD:ACQUISITION:NAME "ACQ\_1"

**FILE:LOAD:SETup:EXECute**

**Function** Loads setting parameters.

**Syntax** FILE:LOAD:SETup:EXECute

**Example** FILE:LOAD:SETUP:EXECUTE

**FILE:LOAD:SETup:NAME**

**Function** Sets the name of the setting parameter file to be loaded.

**Syntax** FILE:LOAD:SETup:NAME <character string>  
<character string>= Up to 8 characters  
can be used, refer to User's manual  
IM701530-01E.

**Example** FILE:LOAD:SETUP:NAME "SETUP\_1"

**FILE:LOAD:TRACe?**

**Function** Queries all the setting values relating to loading P-P waveform data.

**Syntax** FILE:LOAD:TRACe?

**Example** FILE:LOAD:TRACe?→:FILE:LOAD:TRACE:  
TRACE 1

**FILE:LOAD:TRACe:EXECute**

**Function** Loads P-P waveform data. This is an overlap command.

**Syntax** FILE:LOAD:TRACe:EXECute

**Example** FILE:LOAD:TRACE:EXECUTE

**FILE:LOAD:TRACe:NAME**

**Function** Sets the name of the P-P waveform data file to be loaded.

**Syntax** FILE:LOAD:TRACe:NAME <character string>  
<character string>= Up to 8 characters  
can be used, refer to User's manual  
IM701530-01E.

**Example** FILE:LOAD:TRACE:NAME "TRACE\_1"

**FILE:LOAD:TRACe:TRACe**

**Function** Selects the trace No. where the P-P waveform data will be loaded, or queries the current setting.

**Syntax** FILE:LOAD:TRACe:TRACe {<Nrf>}  
FILE:LOAD:TRACe:TRACe?  
{<Nrf>}=1 to 4

**Example** FILE:LOAD:TRACE:TRACE 1  
FILE:LOAD:TRACE:TRACE?→:FILE:LOAD:TRACE:  
TRACE 1

**FILE:PROTeCt:ACQuisition:EXECute**

**Function** Protects acquisition data.

**Syntax** FILE:PROTeCt:ACQuisition:EXECute

**Example** FILE:PROTECT:ACQUISITION:EXECUTE

**FILE:PROTeCt:ACQuisition:NAME**

**Function** Sets the name of the acquisition waveform data file to be protected.

**Syntax** FILE:PROTeCt:ACQuisition:  
NAME <character string>  
<character string>= Up to 8 characters  
can be used, refer to User's manual  
IM701530-01E.

**Example** FILE:PROTECT:ACQUISITION:NAME "ACQ\_1"

**FILE:PROTECT:SETUP:EXECUTE**

**Function** Protects setting parameters.  
**Syntax** FILE:PROTECT:SETUP:EXECUTE  
**Example** FILE:PROTECT:SETUP:EXECUTE

**FILE:PROTECT:SETUP:NAME**

**Function** Sets the name of the setting parameter file to be protected.  
**Syntax** FILE:PROTECT:SETUP:NAME <character string>  
 <character string>= Up to 8 characters can be used, refer to User's manual IM701530-01E.  
**Example** FILE:PROTECT:SETUP:NAME "SETUP\_1"

**FILE:PROTECT:TRACE:EXECUTE**

**Function** Protects P-P waveform data.  
**Syntax** FILE:PROTECT:TRACE:EXECUTE  
**Example** FILE:PROTECT:TRACE:EXECUTE

**FILE:PROTECT:TRACE:NAME**

**Function** Sets the name of the P-P waveform data file to be protected.  
**Syntax** FILE:PROTECT:TRACE:NAME <character string>  
 <character string>= Up to 8 characters can be used, refer to User's manual IM701530-01E.  
**Example** FILE:PROTECT:TRACE:NAME "TRACE\_1"

**FILE:SAVE?**

**Function** Queries all settings relating to saving data.  
**Syntax** FILE:SAVE?  
**Example** FILE:SAVE?→:FILE:SAVE:TRACE:TRACE 1

**FILE:SAVE:ACQUISITION:EXECUTE**

**Function** Saves acquisition waveform data.  
**Syntax** FILE:SAVE:ACQUISITION:EXECUTE  
**Example** FILE:SAVE:ACQUISITION:EXECUTE

**FILE:SAVE:ACQUISITION:NAME**

**Function** Sets the name of the acquisition waveform data file to be saved.  
**Syntax** FILE:SAVE:ACQUISITION:NAME <character string>  
 <character string>= Up to 8 characters can be used, refer to User's manual IM701530-01E.  
**Example** FILE:SAVE:ACQUISITION:NAME "ACQ\_1"

**FILE:SAVE:SETUP:EXECUTE**

**Function** Saves setting parameters.  
**Syntax** FILE:SAVE:SETUP:EXECUTE  
**Example** FILE:SAVE:SETUP:EXECUTE

**FILE:SAVE:SETUP:NAME**

**Function** Sets the name of the setting parameter file to be saved.  
**Syntax** FILE:SAVE:SETUP:NAME <character string>  
 <character string>=Up to 8 characters can be used, refer to User's manual IM701530-01E.  
**Example** FILE:SAVE:SETUP:NAME "SETUP\_1"

**FILE:SAVE:TRACE?**

**Function** Queries all settings relating to saving P-P waveform data.  
**Syntax** FILE:SAVE:TRACE?  
**Example** FILE:SAVE:TRACE?→:FILE:SAVE:TRACE:TRACE 1

**FILE:SAVE:TRACE:EXECUTE**

**Function** Saves P-P waveform data.  
**Syntax** FILE:SAVE:TRACE:EXECUTE  
**Example** FILE:SAVE:TRACE:EXECUTE

**FILE:SAVE:TRACE:NAME**

**Function** Sets the name of the P-P waveform data file to be saved.  
**Syntax** FILE:SAVE:TRACE:NAME <character string>  
 <character string>= Up to 8 characters can be used, refer to User's manual IM701530-01E.  
**Example** FILE:SAVE:TRACE:NAME "TRACE\_1"

**FILE:SAVE:TRACE:TRACE**

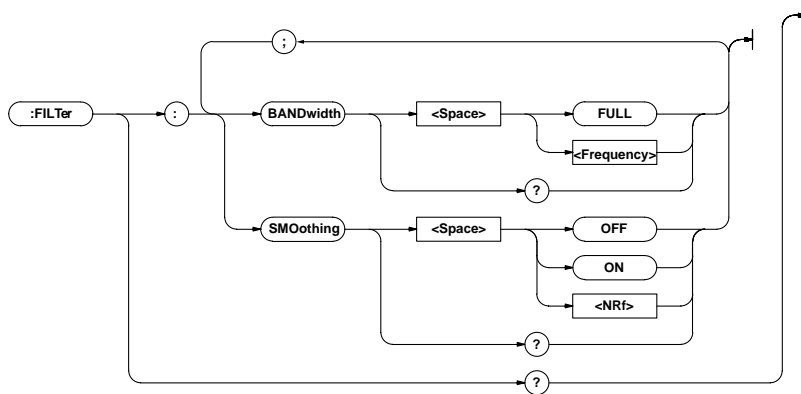
**Function** Selects the trace No. of the P-P waveform data to be saved, or queries the current setting.  
**Syntax** FILE:SAVE:TRACE:TRACE {{{<NRf>}|ALL}  
 FILE:SAVE:TRACE:TRACE?  
 {<NRf>}=1 to 4  
**Example** FILE:SAVE:TRACE:TRACE 1  
 FILE:SAVE:TRACE:TRACE?→:FILE:SAVE:TRACE:TRACE 1

**FILE:SAVE:ACQUISITION:FORM**

**Function** Sets data format for waveform data save, or queries current setting.  
**Syntax** FILE:SAVE:ACQUISITION:FORM {ASCII|BINARY}  
 FILE:SAVE:ACQUISITION:FORM?  
**Example** FILE:SAVE:ACQUISITION:FORM ASCII  
 FILE:SAVE:ACQUISITION:FORM?→:FILE:SAVE:ACQUISITION:FORM ASCII

### 4.11 FILTER Group

The commands in the FILTER group are used to make settings relating to, and inquiries about the input filter. This allows you to make the same settings and inquiries as when using the FILTER key on the front panel.



#### FILTER?

**Function** Queries all the settings relating to the input filter.

**Syntax** FILTER?

**Example** FILTER?→:FILTER:BANDWIDTH FULL;  
SMOOTHING 0

#### FILTER:BANDwidth

**Function** Sets the bandwidth limits, or queries the current setting.

**Syntax** FILTER:BANDwidth {FULL|<frequency>}  
FILTER:BANDwidth?  
<frequency>=20MHZ

**Example** FILTER:BANDWIDTH FULL  
FILTER:BANDWIDTH?→:FILTER:BANDWIDTH FULL

#### FILTER:SMOothing

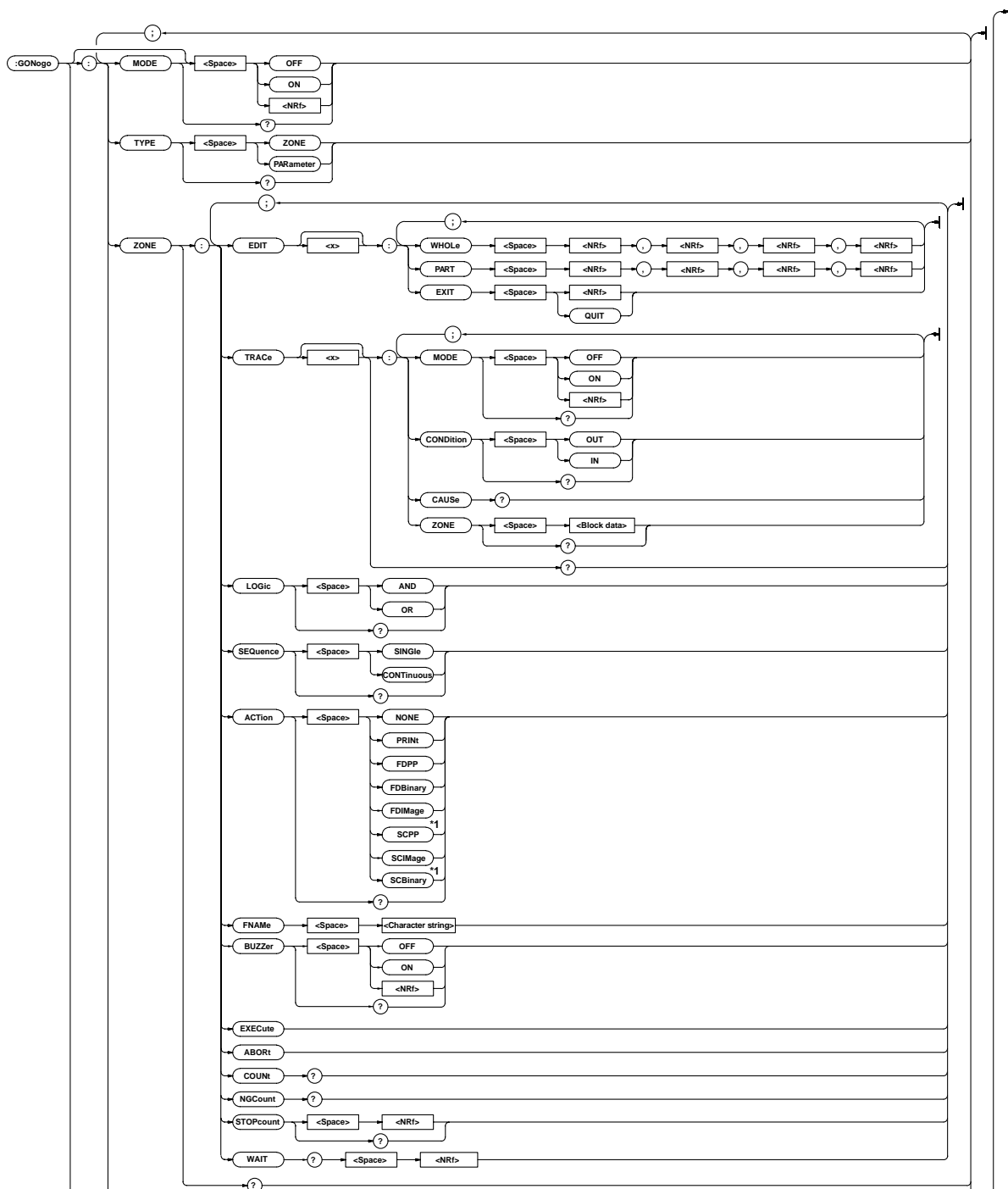
**Function** Turns smoothing ON/OFF, or queries about the current setting.

**Syntax** FILTER:SMOothing {<Boolean>}  
FILTER:SMOothing?

**Example** FILTER:SMOOTHING OFF  
FILTER:SMOOTHING?→:FILTER:SMOOTHING 0

## 4.12 GONogo Group

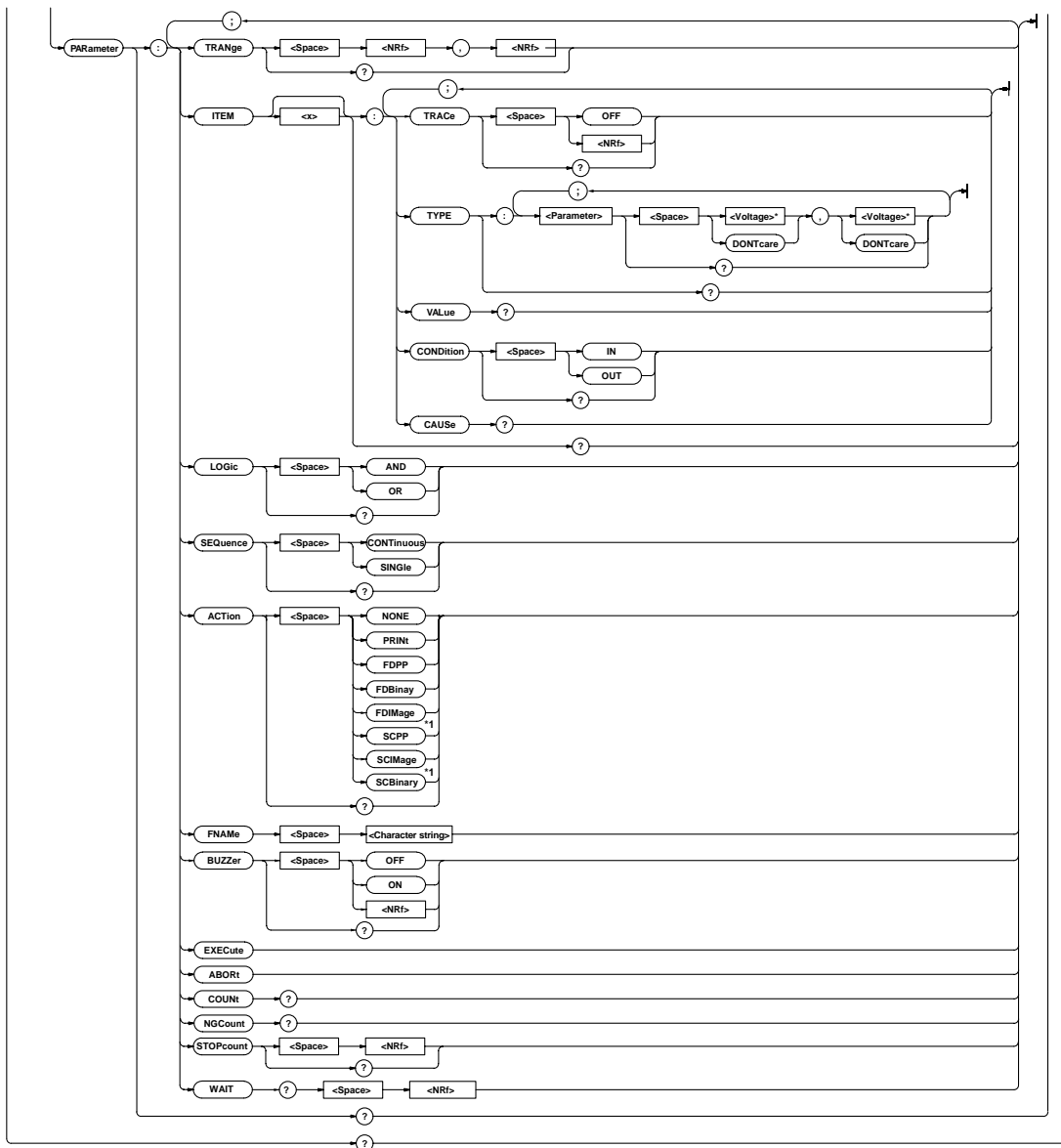
The commands in the GONogo group are used to make settings relating to, and inquiries about GO/NO-GO determination. This allows you to make the same settings and inquiries as when using the GO/NO-GO key on the front panel.



\*1 Available to use the separately sold SCSI interface unit 700930.

\* indicates <voltage> if <Parameter> is AVERAge, HIGH, LOW, MAXimum, MINimum, or RMS (however, {<NRf>} in case of linear scaling); <Time> if <Parameter> is FALL, DELay, NWIDTH, PWIDTH, or RISE; <Frequency> if <Parameter> is FREQUENCY; {<NRf>} if <Parameter> is DUTYcycle, NOVERshoot, PNUMBER, POVERshoot, TY1Integ, TY2Integ, XY1Integ, or XY2Integ

## 4.12 GONogo Group



\*1 Available to use the separately sold SCSI interface unit 700930.

**GONogo?**

**Function** Queries all the GO/NO-GO settings.

**Syntax** GONogo?

**Example** GONOGO?→:GONOGO:MODE 0;TYPE ZONE;ZONE:  
TRACE1:MODE 0;CONDITION OUT;:GONOGO:  
ZONE:TRACE2:MODE 0;CONDITION OUT;:GONOGO:  
ZONE:TRACE3:MODE 0;CONDITION OUT;:GONOGO:  
ZONE:TRACE4:MODE 0;CONDITION OUT;:GONOGO:  
ZONE:LOGIC AND;SEQUENCE SINGLE;  
ACTION NONE;BUZZER 0;:GONOGO:  
PARAMETER:ITEM1:TRACE OFF;TYPE:  
MAXIMUM DONTCARE,DONTCARE;:GONOGO:  
PARAMETER:ITEM1:CONDITION OUT;:GONOGO:  
PARAMETER:ITEM2:TRACE OFF;TYPE:MINIMUM  
DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM2:  
CONDITION OUT;:GONOGO:PARAMETER:ITEM3:  
TRACE OFF;TYPE:RMS DONTCARE,DONTCARE;:  
GONOGO:PARAMETER:ITEM3:CONDITION OUT;:  
GONOGO:PARAMETER:ITEM4:TRACE OFF;TYPE:  
AVERAGE DONTCARE,DONTCARE;:GONOGO:  
PARAMETER:ITEM4:CONDITION OUT;:GONOGO:  
PARAMETER:LOGIC AND;SEQUENCE SINGLE;  
ACTION NONE;TRANGE -5.00,5.00;BUZZER 0

**GONogo[:MODE]**

**Function** Turns GO/NO-GO determination ON/OFF, or queries the current setting.

**Syntax** GONogo[:MODE] {<Boolean>}

**Example** GONOGO:MODE OFF  
GONOGO:MODE?→:GONOGO:MODE 0

**GONogo:PARAMeter?**

**Function** Queries all the settings relating to GO/NO-GO determination which use parameters.

**Syntax** GONogo:PARAMeter?

**Example** GONOGO:PARAMETER?→:GONOGO:PARAMETER:  
ITEM1:TRACE OFF;TYPE:MAXIMUM DONTCARE,  
DONTCARE;:GONOGO:PARAMETER:ITEM1:  
CONDITION OUT;:GONOGO:PARAMETER:ITEM2:  
TRACE OFF;TYPE:MINIMUM DONTCARE,  
DONTCARE;:GONOGO:PARAMETER:ITEM2:  
CONDITION OUT;:GONOGO:PARAMETER:ITEM3:  
TRACE OFF;TYPE:RMS DONTCARE,DONTCARE;:  
GONOGO:PARAMETER:ITEM3:CONDITION OUT;:  
GONOGO:PARAMETER:ITEM4:TRACE OFF;TYPE:  
AVERAGE DONTCARE,DONTCARE;:GONOGO:  
PARAMETER:ITEM4:CONDITIONOUT;:GONOGO:  
PARAMETER:LOGIC AND;SEQUENCE SINGLE;  
ACTION NONE;TRANGE -5.00,5.00;BUZZER 0

**GONogo:PARAMeter:ABORT**

**Function** Aborts parameter GO/NO-GO determination. An error occurs unless both the "GONogo:MODE 1" and "GONogo:TYPE PARAMeter" are set.

**Syntax** GONogo:PARAMeter:ABORT

**Example** GONOGO:PARAMETER:ABORT

**GONogo:PARAMeter:ACTION**

**Function** Sets the action to be taken if the parameter determination result is NO-GO, or queries the current setting.

**Syntax** GONogo:PARAMeter:ACTION {NONE|PRINT|FDPP|FDBinary|FDImage|SCPP|SCBinary|SCImage}  
GONogo:PARAMeter:ACTION?

**Example** GONOGO:PARAMETER:ACTION NONE  
GONOGO:PARAMETER:ACTION?→:GONOGO:  
PARAMETER:ACTION NONE

**Description** • SCPP, SCBinary and SCImage are available to use the separately sold SCSI interface unit 700930.  
• SCPP, SCBinary and SCImage are available only when SCSI is connected.

**GONogo:PARAMeter:BUZZer**

**Function** Sets the ON/OFF setting of the buzzer when NO-GO occurs during parameter GO/NO-GO determination, or queries the current setting.

**Syntax** GONogo:PARAMeter:BUZZer {<Boolean>}

**Example** GONOGO:PARAMETER:BUZZER ON  
GONOGO:PARAMETER:BUZZER?→:GONOGO:  
PARAMETER:BUZZER 1

**GONogo:PARAMeter:COUNT**

**Function** Queries the number of times parameter GO/NO-GO is performed.

**Syntax** GONogo:PARAMeter:COUNT?

**Example** GONOGO:PARAMETER:COUNT?→:GONOGO:  
PARAMETER:COUNT 100

**GONogo:PARAMeter:EXECute**

**Function** Executes parameter GO/NO-GO determination. An error occurs unless both "GONogo:MODE 1" and "GONogo:TYPE PARAMeter" have been set.

**Syntax** GONogo:PARAMeter:EXECute

**Example** GONOGO:PARAMETER:EXECUTE

**GONogo:PARAMeter:FNAME**

**Function** Selects the file name of the NG data to be output.

**Syntax** GONogo:PARAMeter:FNAME <character string>  
<character string>= Up to 4 characters can be used, refer to User's manual IM701530-01E.

**Example** GONogo:PARAMeter:FNAME "NOGO"

**Description** The specified name is used as the common file name, and a serial number is added to the end of the common file automatically.

**GONogo:PARAMeter:ITEM<x>?**

**Function** Queries all the settings relating to the specified waveform parameter for parameter GO/NO-GO determination.

**Syntax** GONogo:PARAMeter:ITEM<x>?  
<x>=1 to 4

**Example** GONOGO:PARAMETER:ITEM1?→:GONOGO:  
PARAMETER:ITEM1:TRACE OFF;TYPE:  
MAXIMUM DONTCARE,DONTCARE;:GONOGO:  
PARAMETER:ITEM1:CONDITION OUT



**GONogo:PARAmeter:ITEM<x>:CAUSE?**

**Function** Queries whether the specified waveform parameter used for parameter GO/NO-GO determination is the cause of failure.

**Syntax** GONogo:PARAmeter:ITEM<x>:CAUSE?  
<x>=1 to 4

**Example** GONOGO:PARAMETER:ITEM1:CAUSE?→:GONOGO:PARAMETER:ITEM1:CAUSE 1

**Description** “1” will be returned if the specified waveform parameter is the cause of failure. “0” will be returned if it is not.

**GONogo:PARAmeter:ITEM<x>:CONDition**

**Function** Sets the criteria for the specified waveform parameter used for parameter GO/NO-GO determination, or queries the current setting.

**Syntax** GONogo:PARAmeter:ITEM<x>:  
CONDition {IN|OUT}  
GONogo:PARAmeter:ITEM<x>:CONDition?  
<x>=1 to 4

**Example** GONOGO:PARAMETER:ITEM1:CONDITION OUT  
GONOGO:PARAMETER:ITEM1:CONDITION?→:  
GONOGO:PARAMETER:ITEM1:CONDITION OUT

**GONogo:PARAmeter:ITEM<x>:TRACe**

**Function** Sets the target waveform (trace) for the specified waveform parameter used for parameter GO/NO-GO determination, or queries the current setting.

**Syntax** GONogo:PARAmeter:ITEM<x>:TRACe {{<NRf>}  
|OFF}  
GONogo:PARAmeter:ITEM<x>:TRACe?  
<x>=1 to 4  
{<NRf>}=1 to 4

**Example** GONOGO:PARAMETER:ITEM1:TRACE 1  
GONOGO:PARAMETER:ITEM1:TRACE?→:  
GONOGO:PARAMETER:ITEM1:TRACE 1

**GONogo:PARAmeter:ITEM<x>:TYPE?**

**Function** Sets the upper/lower limits for the specified parameter used for parameter GO/NO-GO determination, or queries the current setting.

**Syntax** GONogo:PARAmeter:ITEM<x>:TYPE?  
<x>=1 to 4

**Example** GONOGO:PARAMETER:ITEM1:TYPE?→:  
GONOGO:PARAMETER:ITEM1:TYPE:  
MAXIMUM 50E+00, 20E+00

**GONogo:PARAmeter:ITEM<x>:TYPE:****<parameter>**

**Function** Sets the upper/lower limits for the specified measurement item, or queries the current setting.

**Syntax** GONogo:PARAmeter:ITEM<x>:TYPE:<parameter>  
{<{voltage|DONTcare}>,<{voltage|DONTcare}>  
>|<{time|DONTcare}>,<{time|DONTcare}>|<  
{frequency|DONTcare}>,<{frequency|DONTcare}>  
>|<{{<NRf>}|DONTcare}>,<{{<NRf>}|DO  
NTcare}>}

GONogo:PARAmeter:ITEM<x>:TYPE:  
<parameter>?

<parameter>={AVERAge|BWiDth|DElay|  
DUTYcycLe|FALL|FREQUency|HIGH|LOW|  
MAXimum|MINimum|NOVershoot|NWiDth|  
PERiod|PNUMber|POVershoot|PTOPeak|  
PWiDth|RISE|RMS|TY1Integ|TY2Integ|  
XY1Integ|XY2Integ}

<voltage>,<time>,<frequency>,<{<NRf>}  
refer to the User’s Manual IM701530-01E.

**Example** GONOGO:PARAMETER:ITEM1:TYPE:  
AVERAGE 100mV,-100mV  
GONOGO:PARAMETER:ITEM1:TYPE:AVERAGE?→:  
GONOGO:PARAMETER:ITEM1:TYPE:  
AVERAGE 100.0E-03,-100.0E-03

**Description** It is not possible to make an inquiry about parameters other than those specified by this command.

**GONogo:PARAmeter:ITEM<x>:VALue?**

**Function** Queries the measured value of the specified parameter used for parameter GO/NO-GO determination.

**Syntax** GONogo:PARAmeter:ITEM<x>:VALue?  
<x>=1 to 4

**Example** GONOGO:PARAMETER:ITEM1:VALUE?→:GONOGO:  
PARAMETER:ITEM1:VALUE 1.98E-03

**Description** If measurement is not possible because the mode is OFF or for any other reason, “NAN” (non-numeric) will be returned.

**GONogo:PARAmeter:LOGic**

**Function** Sets the determination logic for parameter GO/NO-GO, or queries the current setting.

**Syntax** GONogo:PARAmeter:LOGic {AND|OR}  
GONogo:PARAmeter:LOGic?

**Example** GONOGO:PARAMETER:LOGIC AND  
GONOGO:PARAMETER:LOGIC?→:GONOGO:  
PARAMETER:LOGIC AND

**GONogo:PARAmeter:NGCount?**

**Function** Queries the number of NGs occurring during parameter GO/NO-GO determination.

**Syntax** GONogo:PARAmeter:NGCount?

**Example** GONOGO:PARAMETER:NGCOUNT?→:GONOGO:  
PARAMETER:NGCOUNT 5

**GONogo:PARAmeter:SEQuence**

**Function** Sets the number of times parameter GO/NO-GO determination is performed, or queries the current setting.

**Syntax** GONogo:PARAmeter:SEQuence  
{CONTInuous|SINGle}  
GONogo:PARAmeter:SEQuence?

**Example** GONOGO:PARAMETER:SEQUENCE SINGLE  
GONOGO:PARAMETER:SEQUENCE?→:GONOGO:  
PARAMETER:SEQUENCE SINGLE

**GONogo:PARAmeter:STOPcount**

**Function** Sets the number of times to execute parameter determination, or queries the current setting.

**Syntax** GONogo:PARAmeter:STOPcount {<NRf>}  
GONogo:PARAmeter:STOPcount?  
{<NRf>} = 0 to 65535

**Example** GONOGO:PARAMETER:STOPCOUNT 10  
GONOGO:PARAMETER:STOPCOUNT?→:GONOGO:  
PARAMETER:STOPCOUNT 10

**Description** A count of "0" means that parameter determination is not specified.

**GONogo:PARAmeter:TRANge**

**Function** Sets the parameter GO/NO-GO determination range, or queries the current setting.

**Syntax** GONogo:PARAmeter:TRANge {<NRf>}, {<NRf>}  
GONogo:PARAmeter:TRANge?  
{<NRf>}=-5.00 to 5.00(div, 0.02 step)

**Example** GONOGO:PARAMETER:TRANGE -5.00,5.00  
GONOGO:PARAMETER:TRANGE?→:GONOGO:  
PARAMETER:TRANGE -5.00,5.00

**GONogo:PARAmeter:WAIT?**

**Function** Waits for parameter determination to finish with time-out option.

**Syntax** GONogo:PARAmeter:WAIT? {<NRf>}  
{<NRf>}=0 to 864000 (in units of 100 ms)

**Example** GONOGO:PARAMETER:WAIT? 50 (time-out 5 s)  
→:GONOGO:PARAMETER:WAIT 0

**Description** "1" is returned if it timed out: "0" if finished in time.

**GONogo:TYPE**

**Function** Sets the GO/NO-GO type, or queries the current setting.

**Syntax** GONogo:TYPE {PARAmeter|ZONE}  
GONogo:TYPE?

**Example** GONOGO:TYPE ZONE  
GONOGO:TYPE?→:GONOGO:TYPE ZONE

**GONogo:ZONE?**

**Function** Queries all the settings relating to zone GO/NO-GO determination.

**Syntax** GONogo:ZONE?

**Example** GONOGO:ZONE?→:GONOGO:ZONE:TRACE1:MODE 0;  
CONDITION OUT;:GONOGO:ZONE:TRACE2:MODE 0;  
CONDITION OUT;:GONOGO:ZONE:TRACE3:MODE 0;  
CONDITION OUT;:GONOGO:ZONE:TRACE4:MODE 0;  
CONDITION OUT;:GONOGO:ZONE:LOGIC AND;  
SEQUENCE SINGLE;ACTION NONE

**GONogo:ZONE:ABORt**

**Function** Aborts GO/NO-GO determination. An error occurs unless both "GONogo:MODE 1" and "GONogo:TYPE ZONE" have been set.

**Syntax** GONogo:ZONE:ABORt

**Example** GONOGO:ZONE:ABORT

**GONogo:ZONE:ACTIon**

**Function** Sets the action to be taken if the zone determination result is NG, or queries the current setting.

**Syntax** GONogo:ZONE:ACTIon {NONE|PRINT|FDPP|  
FDBinary|FDImage|SCPP|SCBinary|SCImage}  
GONogo:ZONE:ACTIon?

**Example** GONOGO:ZONE:ACTION NONE  
GONOGO:ZONE:ACTION?→:GONOGO:ZONE:  
ACTION NONE

**Description** • SCPP, SCBinary and SCImage are available to use the separately sold SCSI interface unit 700930.

• SCPP, SCBinary and SCImage are available only when SCSI is connected.

**GONogo:ZONE:BUZZer**

**Function** Sets the ON/OFF setting of the buzzer when NO-GO occurs during zone GO/NO-GO determination, or queries the current setting.

**Syntax** GONogo:ZONE:BUZZer {<Boolean>}  
GONogo:ZONE:BUZZer?

**Example** GONOGO:ZONE:BUZZER ON  
GONOGO:ZONE:BUZZER?→:GONOGO:ZONE:  
BUZZER 1

**GONogo:ZONE:COUNT?**

**Function** Queries the number of times zone GO/NO-GO determination is performed.

**Syntax** GONogo:ZONE:COUNT?

**Example** GONOGO:ZONE:COUNT?→:GONOGO:ZONE:  
COUNT 100

**GONogo:ZONE:EDIT<x>:EXIT**

**Function** Exits from the edit menu for the specified zone.

**Syntax** GONogo:ZONE:EDIT<x>:EXIT {{<NRf>}|QUIT}  
<x>=1 to 8 (where, 5 to 8 are the  
specified zones for the respective  
traces [1 to 4])  
{<NRf>}=1 to 4

**Example** GONOGO:ZONE:EDIT1:EXIT 1

**Description** The zone specified by "GONogo:ZONE:EDIT<x>:PART|WHOLE}" will be canceled if this command is not sent. Except during zone determination, this command will result in an error.

**GONogo:ZONE:EDIT<x>:PART**

**Function** Selects partial edit for the specified zone during zone GO/NO-GO determination.

**Syntax** GONogo:ZONE:EDIT<x>:PART  
{<Nrf>},{<Nrf>},{<Nrf>},{<Nrf>}  
<x>=1 to 8 (where, 5 to 8 are the specified zones for the respective traces [1 to 4])  
{<Nrf>}=-5.00 to 5.00(div, 0.02 step, cursors T1 and T2),  
-8.00 to 8.00(div, 0.02 step: vertical)  
(order of T1, T2, ↑, ↓)

**Example** GONOGO:ZONE:EDIT1:PART -2.50,-2.50,1.00,1.00

**Description** The specified zone will be canceled if edit mode is not terminated using "GONogo:ZONE:EDIT<x>:EXIT" after editing has been completed.

**GONogo:ZONE:EDIT<x>:WHOLE**

**Function** Selects total edit for the specified zone.

**Syntax** GONogo:ZONE:EDIT<x>:WHOLE{<Nrf>},{<Nrf>},{<Nrf>},{<Nrf>}  
<x>=1 to 8 (where, 5 to 8 are the specified zones for the respective traces [1 to 4])  
{<Nrf>}=-5.00 to 5.00(div, 0.02 step: horizontal),  
-8.00 to 8.00(div, 0.02 step: vertical)  
(order of ←, →, ↑, ↓)

**Example** GONOGO:ZONE:EDIT1:WHOLE 1.00,1.00,1.00,1.00

**Description** The specified zone will be canceled if edit mode is not terminated using "GONogo:ZONE:EDIT<x>:EXIT" after editing has been completed.

**GONogo:ZONE:EXECute**

**Function** Executes zone determination. An error occurs unless both "GONogo:MODE 1" and "GONogo:TYPE ZONE" have been set.

**Syntax** GONogo:ZONE:EXECute

**Example** GONOGO:ZONE:EXECUTE

**GONogo:ZONE:FNAME**

**Function** Selects the file name of the NO-GO data to be output.

**Syntax** GONogo:ZONE:FNAME <character string>  
<character string>= Up to 4 characters can be used, refer to User's manual IM701530-01E.

**Example** GONOGO:ZONE:FNAME "NOGO"

**Description** The specified name is used as the common file name, and a serial number (4 char.) is added to the end of the common file name automatically.

**GONogo:ZONE:LOGic**

**Function** Sets the determination logic for zone GO/NO-GO determination, or queries the current settings.

**Syntax** GONogo:ZONE:LOGic {AND|OR}  
GONogo:ZONE:LOGic?

**Example** GONOGO:ZONE:LOGIC AND  
GONOGO:ZONE:LOGIC?→:GONOGO:ZONE:LOGIC AND

**GONogo:ZONE:NGCount?**

**Function** Queries the number of NO-GO's occurring during zone GO/NO-GO determination.

**Syntax** GONogo:ZONE:NGCount?

**Example** GONOGO:ZONE:NGCOUNT?→:GONOGO:ZONE:NGCOUNT 5

**GONogo:ZONE:SEQuence**

**Function** Sets the number of time zones GO/NO-GO determination is performed, or queries the current setting.

**Syntax** GONogo:ZONE:SEQuence {CONTinuous|SINGLE}  
GONogo:ZONE:SEQuence?

**Example** GONOGO:ZONE:SEQUENCE SINGLE  
GONOGO:ZONE:SEQUENCE?→:GONOGO:ZONE:SEQUENCE SINGLE

**GONogo:ZONE:STOPcount**

**Function** Sets the number of times to execute zone determination, or queries the current setting.

**Syntax** GONogo:ZONE:STOPcount {<Nrf>}  
GONogo:ZONE:STOPcount?  
{<Nrf>}=0 to 65535

**Example** GONOGO:ZONE:STOPCOUNT 10  
GONOGO:ZONE:STOPCOUNT?→:GONOGO:ZONE:STOPCOUNT 10

**Description** A count of "0" means that zone determination is not specified.

**GONogo:ZONE:TRACe<x>?**

**Function** Queries all the settings made for the specified zone for zone GO/NO-GO determination.

**Syntax** GONogo:ZONE:TRACe<x>?  
<x>=1 to 4

**Example** GONOGO:ZONE:TRACE1?→:GONOGO:ZONE:TRACE1;MODE 0:CONDITION OUT

**GONogo:ZONE:TRACe<x>:CAUSE?**

**Function** Queries whether the specified trace used for zone GO/NO-GO determination is the cause of failure.

**Syntax** GONogo:ZONE:TRACe<x>:CAUSE?  
<x>=1 to 4

**Example** GONOGO:ZONE:TRACE1:CAUSE?→:GONOGO:ZONE:TRACE1:CAUSE 1

**Description** "1" will be returned if the specified channel is the cause of failure. "0" will be returned if it is not.

**GONogo:ZONE:TRACe<x>:CONDition**

**Function** Sets the criteria for the specified trace used for zone GO/NO-GO determination, or queries the current setting.

**Syntax** GONogo:ZONE:TRACe<x>:CONDition {IN|OUT}  
GONogo:ZONE:TRACe<x>:CONDition?  
<x>=1 to 4

**Example** GONOGO:ZONE:TRACE1:CONDITION OUT  
GONOGO:ZONE:TRACE1:CONDITION?→:GONOGO:  
ZONE:TRACE1:CONDITION OUT

**GONogo:ZONE:TRACe<x>:MODE**

**Function** Selects whether the zone determination result for the specified trace is validated or not, or queries the current setting.

**Syntax** GONogo:ZONE:TRACe<x>:MODE {<Boolean>}  
GONogo:ZONE:TRACe<x>:MODE?  
<x>=1 to 4

**Example** GONOGO:ZONE:TRACE1:MODE OFF  
GONOGO:ZONE:TRACE1:MODE?→:GONOGO:ZONE:  
TRACE1:MODE 0

**GONogo:ZONE:TRACe<x>:ZONE**

**Function** Sets or queries the zone data for each trace.

**Syntax** GONogo:ZONE:TRACe<x>:ZONE #6004020(Data  
byte string)  
GONogo:ZONE:TRACe<x>:ZONE?  
<x>=1 to 4

**Example** GONOGO:ZONE:TRACE1:ZONE #6004020(Data  
byte string)  
GONOGO:ZONE:TRACE1:ZONE?→:GONOGO:ZONE:  
TRACE1:ZONE #6004020(Data byte string)

- Description**
- The <data byte string> is 4020 bytes of zone data.
  - The zone data consists of 8 bits (range from 0 to 255).  
Controller side needs to take special care because the data may contain ASCII codes "0AH" or "00H" which indicate "NL."
  - This command is used to save the zone data from the instrument to the computer, then setting the saved data as zone data to the instrument.
  - After setting the zone data from the computer to the instrument with this command, make sure to set "GONogo:ZONE:TRACe<x>:MODE ON".

**GONogo:ZONE:WAIT?**

**Function** Waits for zone determination to finish with time-out option.

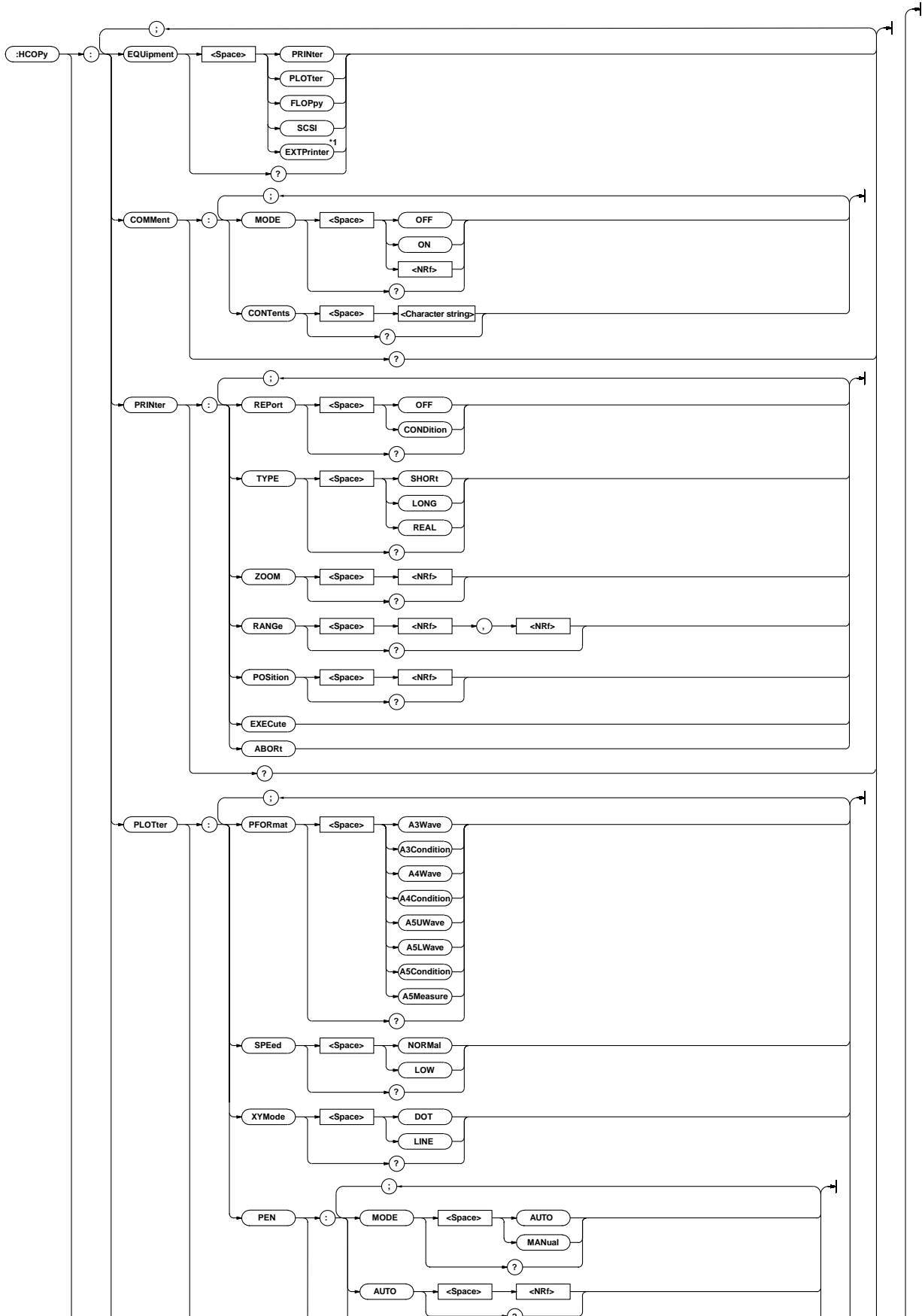
**Syntax** GONogo:ZONE:WAIT? {<NRF>}  
{<NRF>}=0 to 864000 (in units of 100 ms)

**Example** GONOGO:ZONE:WAIT? 50 (time-out 5 s)  
→:GONOGO:ZONE:WAIT 0

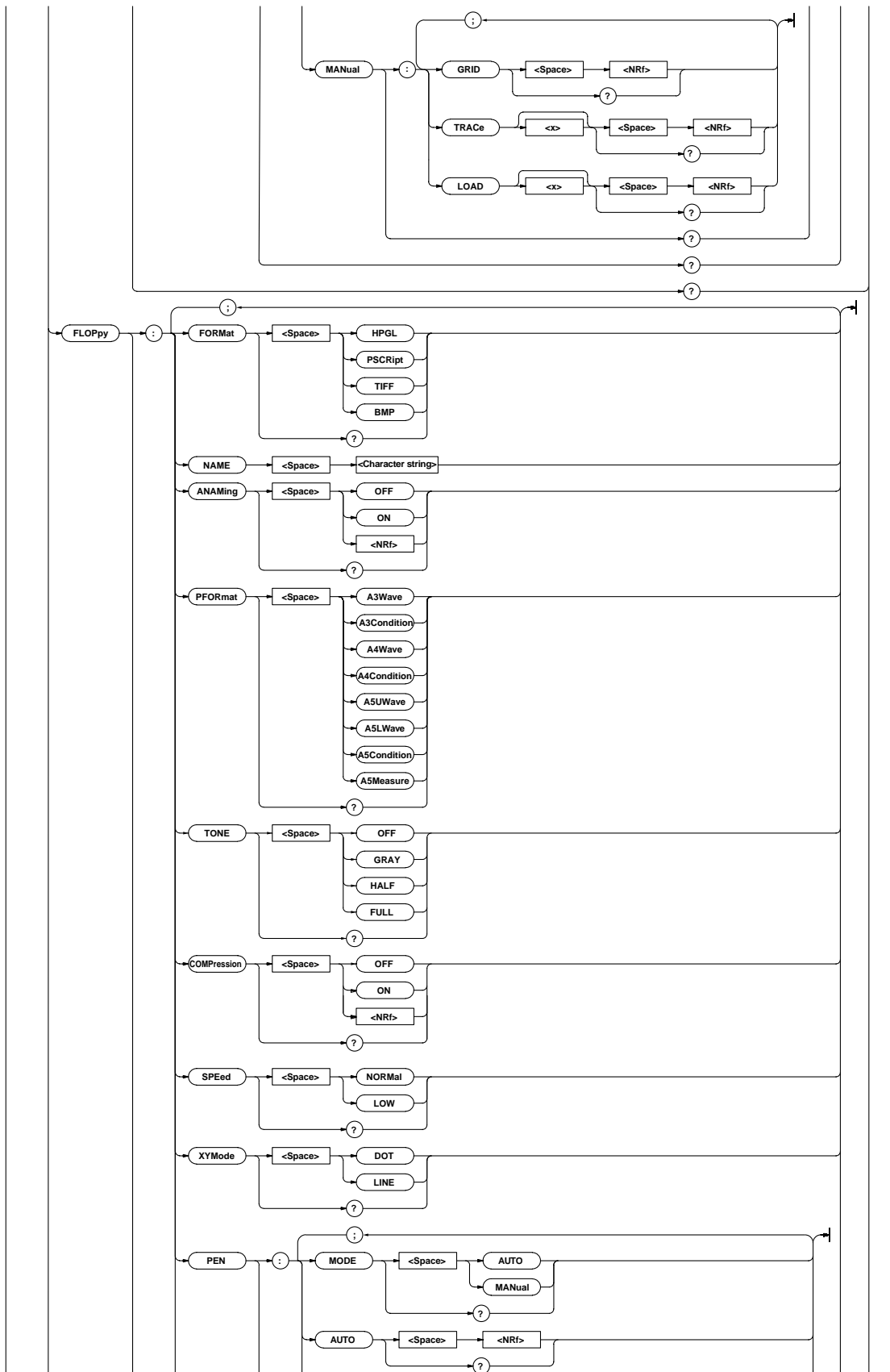
**Description** "1" is returned if it timed out: "0" if finished in time.

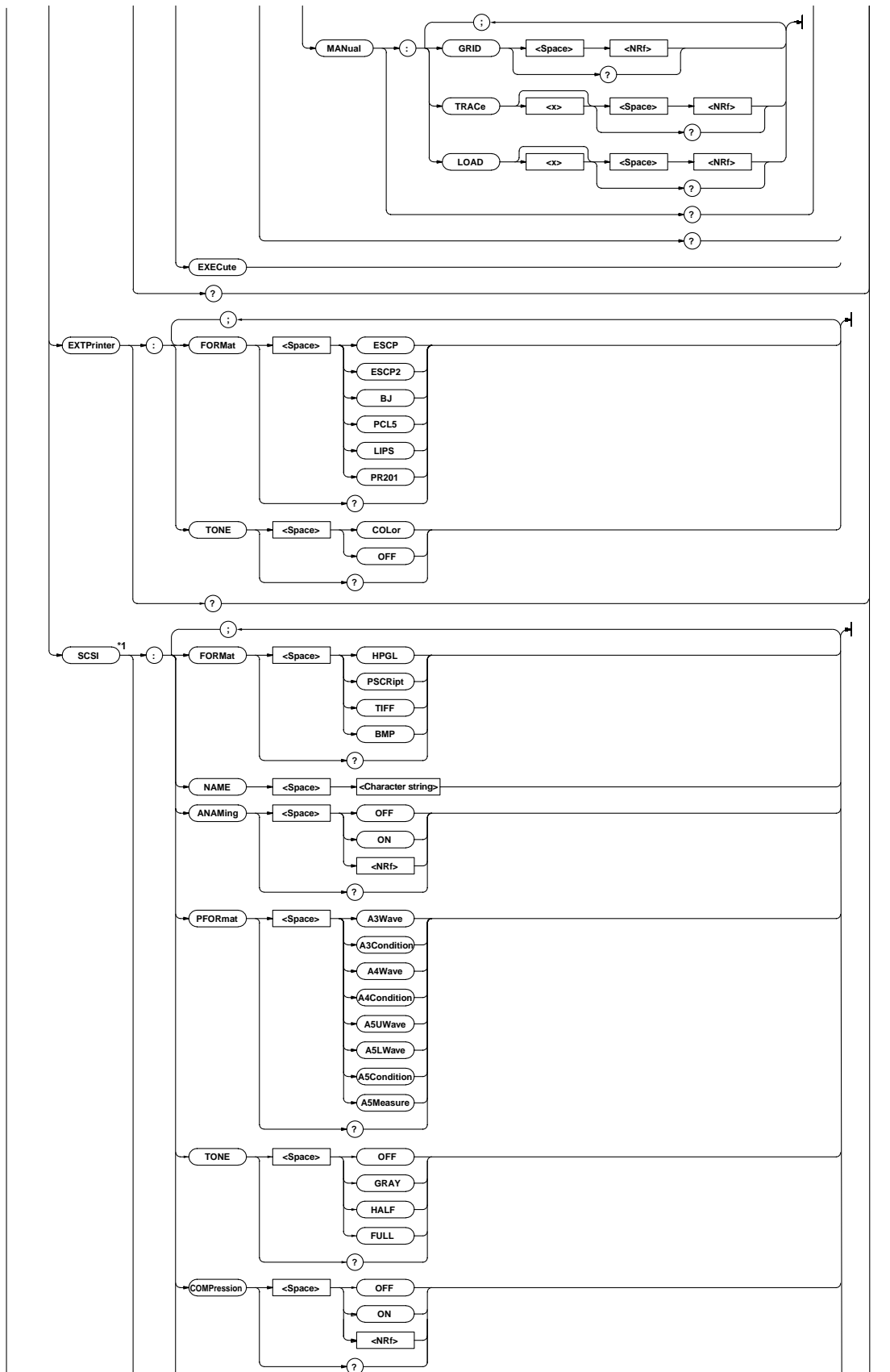
### 4.13 HCOPY Group

The commands in the HCOPY group are used to make settings relating to, and inquiries about screen data output to the optional built-in printer or external HP-GL plotter. You can make the same settings and inquiries as when using the COPY key on the front panel.

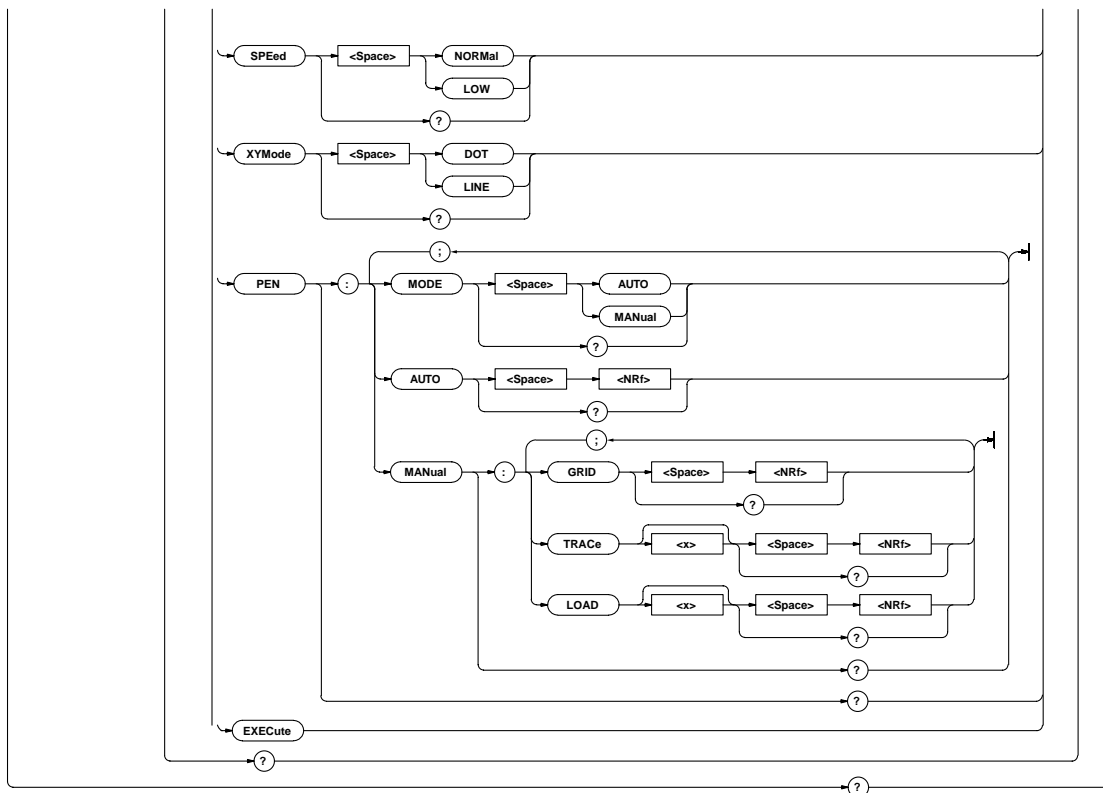


\*1 Available to use the separately sold SCSI interface unit 700930.





\*1 Available to use the separately sold SCSI interface unit 700930..



### HCOpy?

**Function** Queries all the settings relating to output of screen data.

**Syntax** HCOpy?

**Example** HCOpy?→:HCOpy:EQUIPMENT PRINTER;PRINTER:REPORT OFF;TYPE SHORT;RANGE -5.00,5.00;ZOOM 1.0E+00;POSITION 0.0000;:HCOpy:PLOTTER:PFORMAT A4WAVE;SPEED NORMAL;XYMODE DOT;PEN:MODE AUTO;AUTO 5;MANUAL:GRID 1;TRACE1 2;TRACE2 3;TRACE3 4;TRACE4 5;LOAD1 6;LOAD2 7;LOAD3 8;LOAD4 9;:HCOpy:FLOPPY:FORMAT HPGL;ANAMING 0;TONE OFF;COMPRESSION 0;PFORMAT A4WAVE;SPEED NORMAL;XYMODE DOT;PEN:MODE AUTO;AUTO 5;MANUAL:GRID 1;TRACE1 2;TRACE2 3;TRACE3 4;TRACE4 5;LOAD1 6;LOAD2 7;LOAD3 8;LOAD4 9;:HCOpy:EXTPRINTER:FORMAT ESCP;TONE OFF;:HCOpy:COMMENT:MODE 0;CONTENTS ""

### HCOpy:COMMeNt?

**Function** Queries all the settings relating to the comment to be printed on the built-in printer or external plotter.

**Syntax** HCOpy:COMMeNt?

**Example** HCOpy:COMMeNt?→:HCOpy:COMMeNt:MODE 1;CONTENTS "COMMENT"

### HCOpy:COMMeNt:CONTeNts

**Function** Sets the comment to be printed on the built-in printer or external plotter, or queries the current setting.

**Syntax** HCOpy:COMMeNt:CONTeNts <character string>  
HCOpy:COMMeNt:CONTeNts?  
<character string>= Up to 27 characters, refer to User's Manual 701510-01E.

**Example** HCOpy:COMMeNt:CONTeNts "COMMENT"  
HCOpy:COMMeNt:CONTeNts?→:HCOpy:COMMeNt:CONTeNts "COMMENT"

### HCOpy:COMMeNt:MODe

**Function** Turns the printing of comments to the built-in printer or external plotter ON/OFF, or queries the current setting.

**Syntax** HCOpy:COMMeNt:MODe {<Boolean>}  
HCOpy:COMMeNt:MODe?

**Example** HCOpy:COMMeNt:MODe ON  
HCOpy:COMMeNt:MODe?→:HCOpy:COMMeNt:MODe 1

### HCOpy:EQUIPmeNt

**Function** Selects the output media to which the screen data is to be output, or queries the current setting.

**Syntax** HCOpy:EQUIPmeNt {PRINteR|PLOTteR|FLOPPy|EXTPRINteR|SCSI}  
HCOpy:EQUIPmeNt?

**Example** HCOpy:EQUIPmeNt PRINteR  
HCOpy:EQUIPmeNt?→:HCOpy:EQUIPmeNt PRINteR

**Description** • SCSI is available to use the separately sold SCSI interface unit 700930.  
• SCSI is available only when it is connected.



**HCOPY:EXTPrinter?**

**Function** Queries all the settings relating to output an external printer.

**Syntax** HCOpy:EXTPrinter?

**Example** HCOpy:EXTPRINTER?→:HCOpy:EXTPRINTER:  
FORMAT ESCP;TONE OFF

**HCOPY:EXTPrinter:FORMAt**

**Function** Sets the data format for printout to an external printer, or queries the current setting.

**Syntax** HCOpy:EXTPrinter:FORMAt  
{ESCP|ESCP2|BJ|PCL5|LIPS|PR201}  
HCOpy:EXTPrinter:FORMAt?

**Example** HCOpy:EXTPRINTER:FORMAt  
HCOpy:EXTPRINTER:FORMAt?→:HCOpy:  
EXTPRINTER:FORMAt ESCP

**HCOPY:EXTPrinter:TONE**

**Function** Sets the half tone of the external printer output, or queries the current setting.

**Syntax** HCOpy:EXTPrinter:TONE {COLOr|OFF}  
HCOpy:EXTPrinter:TONE?

**Example** HCOpy:EXTPRINTER:TONE COLOr  
HCOpy:EXTPRINTER:TONE?→:HCOpy:  
EXTPRINTER:TONE COLOr

**HCOPY:FLOppy?**

**Function** Queries all the settings relating to saving of screen image data on a floppy disk.

**Syntax** HCOpy:FLOppy?

**Example** HCOpy:FLOPPY?→:HCOpy:FLOPPY:FORMAt HPGL;  
ANAMING 0;TONE OFF;COMPRESSION 0;  
PFORMAt A4WAVE;SPEED NORMAL;XYMODE DOT;  
PEN:MODE AUTO;AUTO 5;MANUAL:GRID 1;  
TRACE1 2;TRACE2 3;TRACE3 4;TRACE4 5;  
LOAD1 6;LOAD2 7;LOAD3 8;LOAD4 9

**HCOPY:FLOppy:ANAMing**

**Function** Sets the auto naming mode for the file name of the screen image data to be output, or queries the current setting.

**Syntax** HCOpy:FLOppy:ANAMing {<BooLear>}  
HCOpy:FLOppy:ANAMing?

**Example** HCOpy:FLOPPY:ANAMING ON  
HCOpy:FLOPPY:ANAMING?→:HCOpy:FLOPPY:  
ANAMING 1

**HCOPY:FLOppy:COMPRESSion**

**Function** Set the compression of the screen image data to output ON/OFF, or queries the current setting.

**Syntax** HCOpy:FLOppy:COMPRESSion {<BooLear>}  
HCOpy:FLOppy:COMPRESSion?

**Example** HCOpy:FLOPPY:COMPRESSION ON  
HCOpy:FLOPPY:COMPRESSION?→:HCOpy:FLOPPY:  
COMPRESSION 1

**HCOPY:FLOppy:EXECute**

**Function** Saves the screen image data.

**Syntax** HCOpy:FLOppy:EXECute

**Example** HCOpy:FLOPPY:EXECUTE

**HCOPY:FLOppy:FORMAt**

**Function** Selects the format of the screen image data to be saved, or queries the current setting.

**Syntax** HCOpy:FLOppy:FORMAt {HPGL|PSCRIPT|  
TIFF|BMP}  
HCOpy:FLOppy:FORMAt?

**Example** HCOpy:FLOPPY:FORMAt TIFF  
HCOpy:FLOPPY:FORMAt?→:HCOpy:FLOPPY:  
FORMAt TIFF

**HCOPY:FLOppy:NAME**

**Function** Selects the name of screen image data file to be output.

**Syntax** HCOpy:FLOppy:NAME <character string>  
<character string>=Up to 8 characters,  
refer to User's Manual IM701530-01E.

**Example** HCOpy:FLOPPY:NAME "IMAGE\_1"

**Description** If the auto naming Function is ON, only the first 5 characters of the specified file name are assigned and a 3-digit No. is added automatically.

**HCOPY:FLOppy:PEN?**

**Function** Queries all the settings relating to the pens of the image data to be output.

**Syntax** HCOpy:FLOppy:PEN?

**Example** HCOpy:FLOPPY:PEN?→:HCOpy:FLOPPY:PEN:  
MODE AUTO;AUTO 5;MANUAL:GRID 1:TRACE1 2;  
TRACE2 3;TRACE3 4;TRACE4 5;LOAD1 6;  
LOAD2 7;LOAD3 8;LOAD4 9

**HCOPY:FLOppy:PEN:AUTO**

**Function** Sets the number of pens of the image data to be output in case pens are assigned automatically, or queries the current setting.

**Syntax** HCOpy:FLOppy:PEN:AUTO {<NRf>}  
HCOpy:FLOppy:PEN:AUTO?  
{<NRf>}=1 to 5

**Example** HCOpy:FLOPPY:PEN:AUTO 5  
HCOpy:FLOPPY:PEN:AUTO?→:HCOpy:FLOPPY:  
PEN:AUTO 5

**HCOPY:FLOppy:PEN:MANual?**

**Function** Queries the number of pens of the image data to be output in case pens are assigned manually.

**Syntax** HCOpy:FLOppy:PEN:MANual?

**Example** HCOpy:FLOPPY:PEN:MANual?→:HCOpy:FLOPPY:  
PEN:MANual:GRID 1:TRACE1 2;TRACE2 3;  
TRACE3 4;TRACE4 5;LOAD1 6;LOAD2 7;  
LOAD3 8;LOAD4 9

**HCOPY:FLOppy:PEN:MANual:GRID**

**Function** Sets the pen Nos. for the graticule, cursors, etc. in case pens are assigned manually, or queries the current setting.

**Syntax** HCOpy:FLOppy:PEN:MANual:GRID {<NRf>}  
HCOpy:FLOppy:PEN:MANual:GRID?  
{<NRf>}=0 to 12

**Example** HCOpy:FLOPPY:PEN:MANual:GRID 1  
HCOpy:FLOPPY:PEN:MANual:GRID?→:HCOpy:  
FLOPPY:PEN:MANual:GRID 1

**HCOPY:FLOppy:PEN:MANual:LOAD<x>**

**Function** Sets the pen Nos. of all loaded traces in case pens are assigned manually, or queries the current setting.

**Syntax** HCOPY:FLOppy:PEN:MANual:LOAD<x> {<NRf>}  
HCOPY:FLOppy:PEN:MANual:LOAD<x>?  
<x>=1 to 4  
{<NRf>}=0 to 12

**Example** HCOPY:FLOppy:PEN:MANual:LOAD1 6  
HCOPY:FLOppy:PEN:MANual:LOAD1?→:HCOPY:  
FLOppy:PEN:MANual:LOAD1 6

**HCOPY:FLOppy:PEN:MANual:TRACe<x>**

**Function** Sets the pen No. for the specified trace in case pens are assigned manually, or queries the current setting.

**Syntax** HCOPY:FLOppy:PEN:MANual:TRACe<x> {<NRf>}  
HCOPY:FLOppy:PEN:MANual:TRACe<x>?  
<x>=1 to 4  
{<NRf>}=0 to 12

**Example** HCOPY:FLOppy:PEN:MANual:TRACE1 2  
HCOPY:FLOppy:PEN:MANual:TRACE1?→:HCOPY:  
FLOppy:PEN:MANual:TRACE1 2

**HCOPY:FLOppy:PEN:MODE**

**Function** Sets the pen assignment method, or queries the current setting.

**Syntax** HCOPY:FLOppy:PEN:MODE {AUTO|MANual}  
HCOPY:FLOppy:PEN:MODE?

**Example** HCOPY:FLOppy:PEN:MODE AUTO  
HCOPY:FLOppy:PEN:MODE?→:HCOPY:FLOppy:  
PEN:MODE AUTO

**HCOPY:FLOppy:PFORmat**

**Function** Sets the output size for the screen image data, or queries the current setting.

**Syntax** HCOPY:FLOppy:PFORmat  
{A3Condition|A3Wave|A4Condition|A4Wave|  
A5 Condition|A5LWave|A5Measure|A5U Wave}  
HCOPY:FLOppy:PFORmat?

**Example** HCOPY:FLOppy:PFORmat A4WAVE  
HCOPY:FLOppy:PFORmat?→:HCOPY:FLOppy:  
PFORmat A4WAVE

**HCOPY:FLOppy:SPEEd**

**Function** Sets the pen speed of the screen image data to be output, or queries the current setting.

**Syntax** HCOPY:FLOppy:SPEEd {NORMal|LOW}  
HCOPY:FLOppy:SPEEd?

**Example** HCOPY:FLOppy:SPEED NORMAL  
HCOPY:FLOppy:SPEED?→:HCOPY:FLOppy:  
SPEED NORMAL

**HCOPY:FLOppy:TONE**

**Function** Set the color tone of the screen image data to output, or queries the current setting.

**Syntax** HCOPY:FLOppy:TONE {OFF|GRAY|HALF|FULL}  
HCOPY:FLOppy:TONE?

**Example** HCOPY:FLOppy:TONE FULL  
HCOPY:FLOppy:TONE?→:HCOPY:FLOppy:  
TONE FULL

**HCOPY:FLOppy:XYMode**

**Function** Selects the plot method for X-Y waveforms, or queries the current setting.

**Syntax** HCOPY:FLOppy:XYMode {DOT|LINE}  
HCOPY:FLOppy:XYMode?

**Example** HCOPY:FLOppy:XYMODE LINE  
HCOPY:FLOppy:XYMODE?→:HCOPY:FLOppy:  
XYMODE LINE

**Description** It can be selected irrespective of whether sampling points are connected by dots (DOT) or lines (LINE).

**HCOPY:PLOTter?**

**Function** Queries all the settings relating to output to an HP-GL plotter.

**Syntax** HCOPY:PLOTter?

**Example** HCOPY:PLOTter?→:HCOPY:PLOTter:  
PFORMAT A4WAVE;SPEED NORMAL;XYMODE 0;PEN:  
MODE AUTO;AUTO 5;MANUAL:GRID 1;TRACE1 2;  
TRACE2 3;TRACE3 4;TRACE4 5;LOAD1 6;  
LOAD2 7;LOAD3 8;LOAD4 9

**HCOPY:PLOTter:PEN?**

**Function** Queries all the settings relating to the plotter's pen.

**Syntax** HCOPY:PLOTter:PEN?

**Example** HCOPY:PLOTter:PEN?→:HCOPY:PLOTter:PEN:  
MODE AUTO;AUTO 5;MANUAL:GRID 1;TRACE1 2;  
TRACE2 3;TRACE3 4;TRACE4 5;LOAD1 6;  
LOAD2 7;LOAD3 8;LOAD4 9

**HCOPY:PLOTter:PEN:AUTO**

**Function** Sets the number of pens in case pens are assigned automatically, or queries the current setting.

**Syntax** HCOPY:PLOTter:PEN:AUTO {<NRf>}  
HCOPY:PLOTter:PEN:AUTO?  
{<NRf>}=1 to 5

**Example** HCOPY:PLOTter:PEN:AUTO 5  
HCOPY:PLOTter:PEN:AUTO?→:HCOPY:PLOTter:  
PEN:AUTO 5

**HCOPY:PLOTter:PEN:MANual?**

**Function** Queries the number of pens in case pens are assigned manually.

**Syntax** HCOPY:PLOTter:PEN:MANual?

**Example** HCOPY:PLOTter:PEN:MANual?→:HCOPY:  
PLOTter:PEN:MANual:GRID 1;TRACE1 2;  
TRACE2 3;TRACE3 4;TRACE4 5;LOAD1 6;  
LOAD2 7;LOAD3 8;LOAD4 9

**HCOPY:PLOTter:PEN:MANual:GRID**

**Function** Sets the pen Nos. for the graticule, cursors, etc. in case pens are assigned manually, or queries the current setting.

**Syntax** HCOPY:PLOTter:PEN:MANual:GRID {<NRf>}  
HCOPY:PLOTter:PEN:MANual:GRID?  
{<NRf>}=0 to 12

**Example** HCOPY:PLOTter:PEN:MANual:GRID 1  
HCOPY:PLOTter:PEN:MANual:GRID?→:HCOPY:  
PLOTter:PEN:MANual:GRID 1

**HCOpy:PLOTter:PEN:MANual:LOAD<x>**

**Function** Sets the plotter pen No. for the specified loaded trace in case pens are assigned manually, or queries the current setting.

**Syntax** HCOpy:PLOTter:PEN:MANual:LOAD<x> {<NRf>}  
HCOpy:PLOTter:PEN:MANual:LOAD<x>?  
<x>=1 to 4  
{<NRf>}=0 to 12

**Example** HCOpy:PLOTter:PEN:MANual:LOAD1 6  
HCOpy:PLOTter:PEN:MANual:LOAD1?→:HCOpy:  
PLOTter:PEN:MANual:LOAD1 6

**HCOpy:PLOTter:PEN:MANual:TRACe<x>**

**Function** Sets the plotter pen No. for the specified loaded trace in case pens are assigned manually, or queries the current setting.

**Syntax** HCOpy:PLOTter:PEN:MANual:TRACe<x> {<NRf>}  
HCOpy:PLOTter:PEN:MANual:TRACe<x>?  
<x>=1 to 4  
{<NRf>}=0 to 12

**Example** HCOpy:PLOTter:PEN:MANual:TRACE1 2  
HCOpy:PLOTter:PEN:MANual:TRACE1?→:HCOpy:  
PLOTter:PEN:MANual:TRACE1 2

**HCOpy:PLOTter:PEN:MODE**

**Function** Sets the plotter pen assignment method, or queries the current setting.

**Syntax** HCOpy:PLOTter:PEN:MODE {AUTO|MANual}  
HCOpy:PLOTter:PEN:MODE?

**Example** HCOpy:PLOTter:PEN:MODE AUTO  
HCOpy:PLOTter:PEN:MODE?→:HCOpy:PLOTter:  
PEN:MODE AUTO

**HCOpy:PLOTter:PFORmat**

**Function** Sets the output size to the plotter, or queries the current setting.

**Syntax** HCOpy:PLOTter:PFORmat {A3Condition|  
A3Wave|A4Condition|A4Wave|A5Condition|  
A5Wave|A5Measure|A5UWave}  
HCOpy:PLOTter:PFORmat?

**Example** HCOpy:PLOTter:PFORmat A4WAVE  
HCOpy:PLOTter:PFORmat?→:HCOpy:PLOTter:  
PFORmat A4WAVE

**HCOpy:PLOTter:SPEEd**

**Function** Sets the plotter pen speed, or queries the current setting.

**Syntax** HCOpy:PLOTter:SPEEd {NORMal|LOW}  
HCOpy:PLOTter:SPEEd?

**Example** HCOpy:PLOTter:SPEED NORMAL  
HCOpy:PLOTter:SPEED?→:HCOpy:PLOTter:  
SPEED NORMAL

**HCOpy:PLOTter:XYMode**

**Function** Selects the plot method for X-Y waveforms, or queries the current setting.

**Syntax** HCOpy:PLOTter:XYMode {DOT|LINE}  
HCOpy:PLOTter:XYMode?

**Example** HCOpy:PLOTter:XYMODE LINE  
HCOpy:PLOTter:XYMODE?→:HCOpy:PLOTter:  
XYMODE LINE

**Description** It can be selected irrespective of whether sampling points are connected by dots (DOT) or lines (LINE).

**HCOpy:PRINter?**

**Function** Queries all the settings relating to output to the built-in printer.

**Syntax** HCOpy:PRINter?

**Example** HCOpy:PRINter?→:HCOpy:PRINter:  
REPORT OFF;TYPE SHORT;RANGE -5.00,5.00;  
ZOOM 1.0E+00;POSITION 0.0000

**HCOpy:PRINter:ABORt**

**Function** Aborts print-out to the built-in printer.

**Syntax** HCOpy:PRINter:ABORt

**Example** HCOpy:PRINter:ABORt

**HCOpy:PRINter:EXECute**

**Function** Prints out to the built-in printer. This is an overlap command.

**Syntax** HCOpy:PRINter:EXECute

**Example** HCOpy:PRINter:EXECUTE

**HCOpy:PRINter:POSition**

**Function** Sets the position of the expansion box when printing out a long copy to the built-in printer, or queries the current setting.

**Syntax** HCOpy:PRINter:POSition {<NRf>}  
HCOpy:PRINter:POSition?  
{<NRf>}=-5.0000 to 5.0000(div, 0.0001  
step)

**Example** HCOpy:PRINter:POSITION 1.0000  
HCOpy:PRINter:POSITION?→:HCOpy:  
PRINter:POSITION 1.0000

**HCOpy:PRINter:RANGE**

**Function** Sets the output range for a long copy to the built-in printer, or queries the current setting.

**Syntax** HCOpy:PRINter:RANGE {<NRf>},{<NRf>}  
HCOpy:PRINter:RANGE?  
{<NRf>}=-5.00 to 5.00

**Example** HCOpy:PRINter:RANGE -2.50,2.50  
HCOpy:PRINter:RANGE?→:HCOpy:PRINter:  
RANGE -2.50,2.50

**HCOpy:PRINter:REPORt**

**Function** Selects whether a report is to be output to the built-in printer, or queries the current setting.

**Syntax** HCOpy:PRINter:REPORt {CONDition|OFF}  
HCOpy:PRINter:REPORt?

**Example** HCOpy:PRINter:REPORT CONDITION  
HCOpy:PRINter:REPORT?→:HCOpy:PRINter:  
REPORT CONDITION

**HCOPY:PRINter:TYPE**

**Function** Selects short copy or long copy when outputting to the built-in printer, or queries the current setting.

**Syntax** HCOPY:PRINter:TYPE {LONG|SHORT|REAL}  
HCOPY:PRINter:TYPE?

**Example** HCOPY:PRINter:TYPE SHORT  
HCOPY:PRINter:TYPE?→:HCOPY:PRINter:  
TYPE SHORT

**HCOPY:PRINter:ZOOM**

**Function** Sets the zoom rate used for a long copy to the built-in printer, or queries the current setting.

**Syntax** HCOPY:PRINter:ZOOM {<NRF>}  
HCOPY:PRINter:ZOOM?  
{<NRF>}=1 to 10000 (refer to User's  
Manual IM701530-01E)

**Example** HCOPY:PRINter:ZOOM 1  
HCOPY:PRINter:ZOOM?→:HCOPY:PRINter:  
ZOOM 1

**HCOPY:SCSI?**

**Function** Queries all the settings relating to saving of screen image data on a SCSI device.

**Syntax** HCOPY:SCSI?

**Example** HCOPY:SCSI?→:HCOPY:SCSI:FORMAT HPGL;  
ANAMING 0;TONE OFF;COMPRESSION 0;  
PFORMAT A4WAVE;SPEED NORMAL;XYMODE DOT;  
PEN:MODE AUTO;AUTO 5;MANUAL:GRID 1;  
TRACE1 2;TRACE2 3;TRACE3 4;TRACE4 5;  
LOAD1 6;LOAD2 7;LOAD3 8;LOAD4 9

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

**HCOPY:SCSI:ANAMing**

**Function** Sets the auto naming mode for the file name of the screen image data to output to SCSI, or queries the current setting.

**Syntax** HCOPY:SCSI:ANAMing {<Boolean>}  
HCOPY:SCSI:ANAMing?

**Example** HCOPY:SCSI:ANAMING ON  
HCOPY:SCSI:ANAMING?→:HCOPY:SCSI:  
ANAMING ON

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

**HCOPY:SCSI:COMPression**

**Function** Set the compression of the screen image data to output ON/OFF, or queries the current setting.

**Syntax** HCOPY:SCSI:COMPression {<Boolean>}  
HCOPY:SCSI:COMPression?

**Example** HCOPY:SCSI:COMPRESSION ON  
HCOPY:SCSI:COMPRESSION?→:HCOPY:SCSI:  
COMPRESSION 1

**HCOPY:SCSI:EXECute**

**Function** Saves the screen image data to SCSI.

**Syntax** HCOPY:SCSI:EXECute

**Example** HCOPY:SCSI:EXECUTE

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

**HCOPY:SCSI:FORMat**

**Function** Selects the format of the screen image data saved to SCSI, or queries the current setting.

**Syntax** HCOPY:SCSI:FORMat {HPGL|PSCRIPT|TIFF|BMP}  
HCOPY:SCSI:FORMat?

**Example** HCOPY:SCSI:FORMAT TIFF  
HCOPY:SCSI:FORMAT?→:HCOPY:SCSI:  
FORMAT TIFF

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

**HCOPY:SCSI:NAME**

**Function** Selects the name of the screen image data file to output to SCSI.

**Syntax** HCOPY:SCSI:NAME <character string>  
<character string>=Up to 8 characters

**Example** HCOPY:SCSI:NAME "IMAGE\_1"

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• If the auto naming function is ON, only the first 5 characters of the specified file name are assigned and a 3-digits No. is added automatically.  
• This results in error if SCSI is not connected.

**HCOPY:SCSI:PEN?**

**Function** Queries all the settings relating to the pens of the image data to output to SCSI.

**Syntax** HCOPY:SCSI:PEN?

**Example** HCOPY:SCSI:PEN?→:HCOPY:SCSI:PEN:  
MODE AUTO;AUTO 5;MANUAL:GRID 1;TRACE1 2;  
TRACE2 3;TRACE3 4;TRACE4 5;LOAD1 6;  
LOAD2 7;LOAD3 8;LOAD4 9

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

**HCOPY:SCSI:PEN:AUTO**

**Function** Sets the number of pens of the image data to output to SCSI when pens are assigned automatically, or queries the current setting.

**Syntax** HCOPY:SCSI:PEN:AUTO {<NRF>}  
HCOPY:SCSI:PEN:AUTO?  
{<NRF>}=1 to 5

**Example** HCOPY:SCSI:PEN:AUTO 5  
HCOPY:SCSI:PEN:AUTO?→:HCOPY:SCSI:PEN:  
AUTO 5

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

**HCOpy:SCSI:PEN:MANual?**

**Function** Queries all the settings relating the number of pens of the image data to output to SCSI when pens are assigned manually.

**Syntax** HCOpy:SCSI:PEN:MANual?

**Example** HCOpy:SCSI:PEN:MANual?→:HCOpy:SCSI:PEN:MANUAL:GRID 1;TRACE1 2;TRACE2 3;TRACE3 4;TRACE4 5;LOAD1 6;LOAD2 7;LOAD3 8;LOAD4 9

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

**HCOpy:SCSI:PEN:MANual:GRID**

**Function** Sets the pen Nos. for the graticule, cursor, etc. when pens used to output screen image data to SCSI are assigned manually, or queries the current setting.

**Syntax** HCOpy:SCSI:PEN:MANual:GRID {<Nrf>}  
HCOpy:SCSI:PEN:MANual:GRID?  
{<Nrf>}=0 to 12

**Example** HCOpy:SCSI:PEN:MANUAL:GRID 1  
HCOpy:SCSI:PEN:MANUAL:GRID?→:HCOpy:SCSI:PEN:MANUAL:GRID 1

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

**HCOpy:SCSI:PEN:MANual:LOAD<x>**

**Function** Sets the pen Nos. of all loaded traces when pens used to output screen image data to SCSI are assigned manually, or queries the current setting.

**Syntax** HCOpy:SCSI:PEN:MANual:LOAD<x> {<Nrf>}  
HCOpy:SCSI:PEN:MANual:LOAD<x>?  
<x>=1 to 4  
{<Nrf>}=0 to 12

**Example** HCOpy:SCSI:PEN:MANUAL:LOAD1 6  
HCOpy:SCSI:PEN:MANUAL:LOAD1?→:HCOpy:SCSI:PEN:MANUAL:LOAD1 6

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

**HCOpy:SCSI:PEN:MANual:TRACe<x>**

**Function** Sets the pen No. for the specified trace when pens used to output screen image data to SCSI are assigned manually, or queries the current setting.

**Syntax** HCOpy:SCSI:PEN:MANual:TRACe<x> {<Nrf>}  
HCOpy:SCSI:PEN:MANual:TRACe<x>?  
<x>=1 to 4  
{<Nrf>}=0 to 12

**Example** HCOpy:SCSI:PEN:MANUAL:TRACE1 2  
HCOpy:SCSI:PEN:MANUAL:TRACE1?→:HCOpy:SCSI:PEN:MANUAL:TRACE1 2

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

**HCOpy:SCSI:PEN:MODE**

**Function** Sets the assignment method for the pens used to output screen image data to SCSI, or queries the current setting.

**Syntax** HCOpy:SCSI:PEN:MODE {AUTO|MANual}  
HCOpy:SCSI:PEN:MODE?

**Example** HCOpy:SCSI:PEN:MODE AUTO  
HCOpy:SCSI:PEN:MODE?→:HCOpy:SCSI:PEN:MODE AUTO

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

**HCOpy:SCSI:PFORmat**

**Function** Sets the output size of the screen image data output to SCSI, or queries the current setting.

**Syntax** HCOpy:SCSI:PFORmat {A3Wave|A3Condition|A4Wave|A4Condition|A5UWave|A5LWave|A5Condition|A5Measure}  
HCOpy:SCSI:PFORmat?

**Example** HCOpy:SCSI:PFORmat A4WAVE  
HCOpy:SCSI:PFORmat?→:HCOpy:SCSI:PFORmat A4WAVE

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

**HCOpy:SCSI:SPEEd**

**Function** Sets the pen speed of the screen image data to output to SCSI, or queries the current setting.

**Syntax** HCOpy:SCSI:SPEEd {NORMAL|LOW}  
HCOpy:SCSI:SPEEd?

**Example** HCOpy:SCSI:SPEED NORMAL  
HCOpy:SCSI:SPEED?→:HCOpy:SCSI:SPEED NORMAL

**Description** • Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

**HCOpy:SCSI:TONE**

**Function** Set the color tone of the screen image data to output, or queries the current setting.

**Syntax** HCOpy:SCSI:TONE {OFF|GRAY|HALF|FULL}  
HCOpy:SCSI:TONE?

**Example** HCOpy:SCSI:TONE FULL  
HCOpy:SCSI:TONE?→:HCOpy:SCSI:TONE FULL

**HCOpy:SCSI:XYMode**

**Function** Selects the plot method for X-Y waveforms of the screen image data to output to SCSI, or queries the current setting.

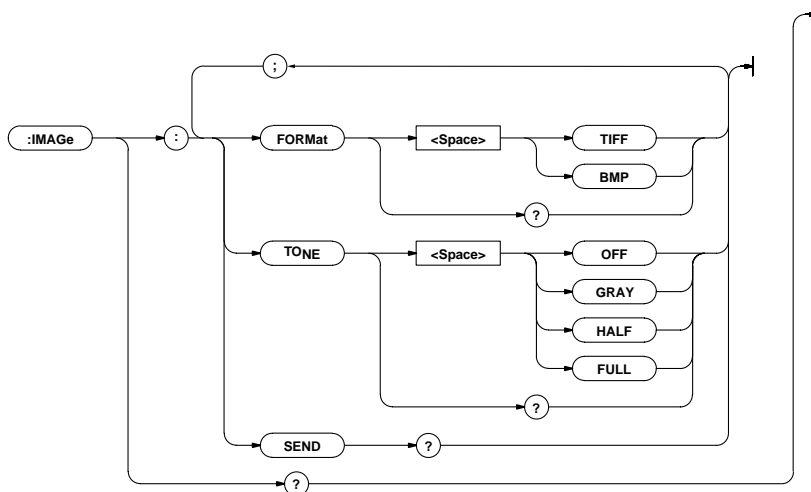
**Syntax** HCOpy:SCSI:XYMode {DOT|LINE}  
HCOpy:SCSI:XYMode?

**Example** HCOpy:SCSI:XYMODE LINE  
HCOpy:SCSI:XYMODE?→:HCOpy:SCSI:XYMODE LINE

**Description** • It can be selected irrespective of whether sampling points are connected by dots (DOT) or lines (LINE).  
• Available to use the separately sold SCSI interface unit 700930.  
• This results in error if SCSI is not connected.

## 4.14 IMAGE Group

Use this group to set or query the screen-image data output settings. There are no corresponding front-panel keys for these operations.



### IMAGE?

**Function** Queries all screen-image data output settings.

**Syntax** IMAGE?

**Example** IMAGE?→:IMAGE:FORMAT TIFF;TONE OFF

### IMAGE:FORMat

**Function** Sets the screen-image data output format, or queries the current setting.

**Syntax** IMAGE:FORMat {TIFF|BMP}

IMAGE:FORMat?

**Example** IMAGE:FORMAT TIFF

IMAGE:FORMAT?→:IMAGE:FORMAT TIFF

### IMAGE:SEND?

**Function** Queries screen-image data.

**Syntax** IMAGE:SEND?

**Example** IMAGE:SEND?→#6(bytes of 6digits)(Data byte string)(Block data)

**Description** Number of bytes in <block data> is  $\{(2+\{6+(\text{Number of data}+\{1(\text{del imiter})\})\})\}$ .  
For information about block data, refer to page 3-6.

### IMAGE:TONE

**Function** Set the color tone of the screen image data to output, or queries the current setting.

**Syntax** IMAGE:TONE {OFF|GRAY|HALF|FULL}

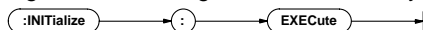
IMAGE:TONE?

**Example** IMAGE:TONE FULL

IMAGE:TONE?→:IMAGE:TONE FULL

## 4.15 INITIALize Group

The INITIALize command is used for making settings relating to initialization. This allows you to make the same settings as when using the INITIALIZE key on the front panel.



### INITialize:EXECute

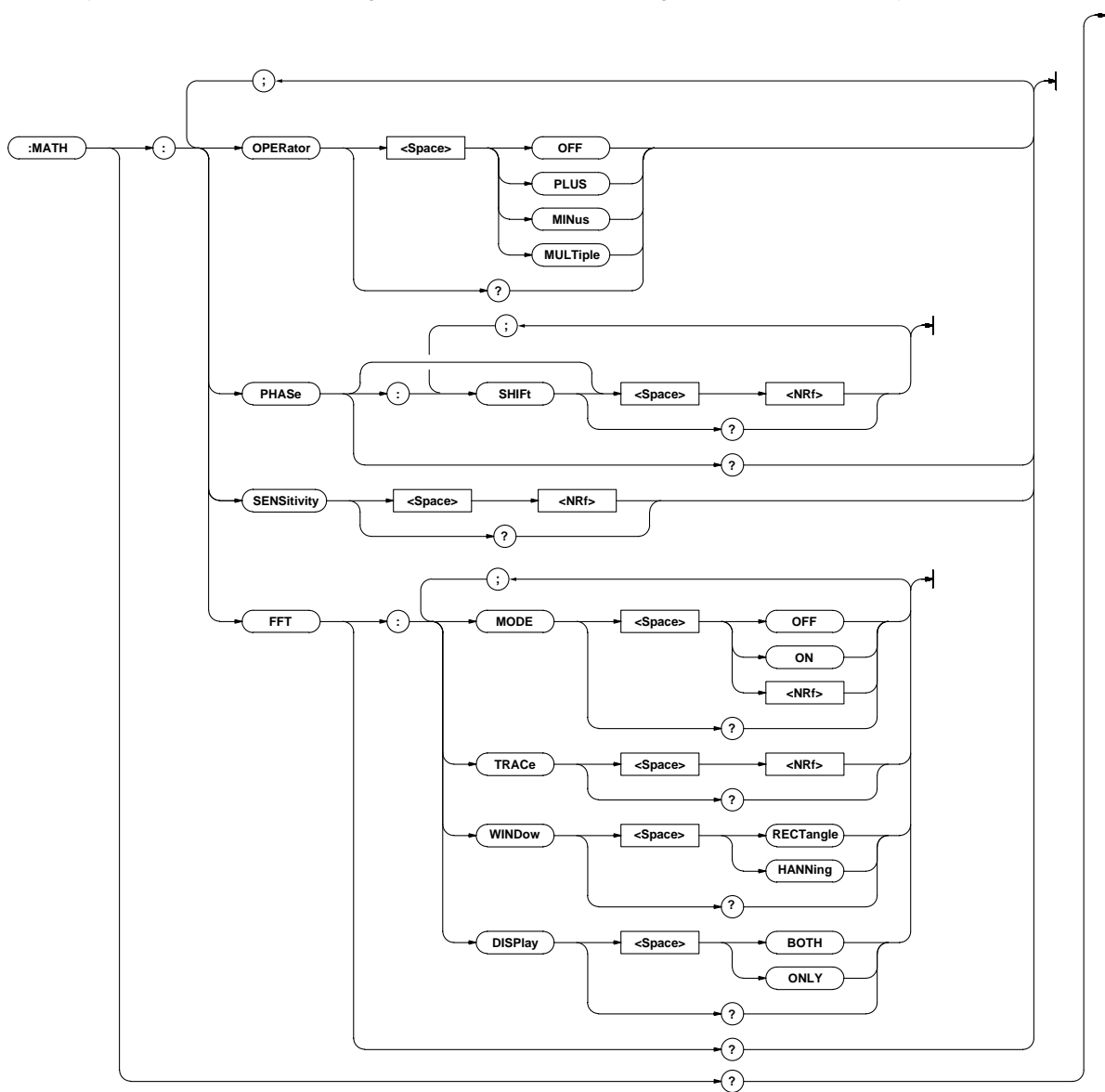
**Function** Executes initialization.

**Syntax** INITialize:EXECute

**Example** INITIALIZE:EXECUTE

### 4.16 MATH Group

The commands in the MATH group are used to make settings relating to, and inquiries about waveform computation. This allows you to make the same settings and inquiries as when using the MATH and FFT key on the front panel.



#### MATH?

**Function** Queries all the settings relating to waveform computation.

**Syntax** MATH?

**Example** MATH?→:MATH:OPERATOR OFF;PHASE:SHIFT 0.0500;:MATH:SENSITIVITY 2.00;FFT:MODE 0;WINDOW HANNING;TRACE 1;DISPLAY BOTH

#### MATH:FFT?

**Function** Queries all the settings relating to FFT computation.

**Syntax** MATH:FFT?

**Example** MATH:FFT?→:MATH:FFT:MODE OFF;WINDOW HANNING;TRACE 1;DISPLAY BOTH

#### MATH:FFT:DISPlay

**Function** Sets the FFT waveform to be displayed, or queries the current setting.

**Syntax** MATH:FFT:DISPlay {BOTH|ONLY}  
MATH:FFT:DISPlay?

**Example** MATH:FFT:DISPLAY BOTH  
MATH:FFT:DISPLAY?→:MATH:FFT:DISPLAY BOTH

#### MATH:FFT:MODE

**Function** Sets the FFT waveform display ON/OFF, or queries the current setting.

**Syntax** MATH:FFT:MODE {<Boolean>}  
MATH:FFT:MODE?

**Example** MATH:FFT:MODE ON  
MATH:FFT:MODE?→:MATH:FFT:MODE 1

**MATH:FFT:TRACe**

**Function** Sets the trace for FFT measurement, or queries the current setting.

**Syntax** MATH:FFT:TRACe {<NRf>}  
MATH:FFT:TRACe?  
{<NRf>}=1 to 4

**Example** MATH:FFT:TRACE 1  
MATH:FFT:TRACE?→:MATH:FFT:TRACE 1

**MATH:FFT:WINDow**

**Function** Sets the FFT window Function, or queries the current setting.

**Syntax** MATH:FFT:WINDow {RECTangle|HANNing}  
MATH:FFT:WINDow?

**Example** MATH:FFT:WINDOW HANNING  
MATH:FFT:WINDOW?→:MATH:FFT:  
WINDOW HANNING

**MATH:OPERator**

**Function** Sets the computation contents.

**Syntax** MATH:OPERator {MINus|MULTiPle|OFF|PLUS}  
MATH:OPERator?

**Example** MATH:OPERATOR OFF  
MATH:OPERATOR?→:MATH:OPERATOR OFF

**MATH:PHASe?**

**Function** Queries all the settings relating to phase shift.

**Syntax** MATH:PHASe?

**Example** MATH:PHASE?→:MATH:PHASE:SHIFT 0.0000

**MATH:PHASe[:SHIFt]**

**Function** Sets the desired phase shift, or queries the current setting.

**Syntax** MATH:PHASe[:SHIFt] {<NRf>}  
MATH:PHASe[:SHIFt]?  
{<NRf>}=-5.0000 to 5.0000 (div, 0.0001  
step)

**Example** MATH:PHASE:SHIFT 0  
MATH:PHASE:SHIFT?→:MATH:PHASE:  
SHIFT 0.0000

**MATH:SENSitivity**

**Function** Sets the display sensitivity of the vertical axis, or queries the current setting.

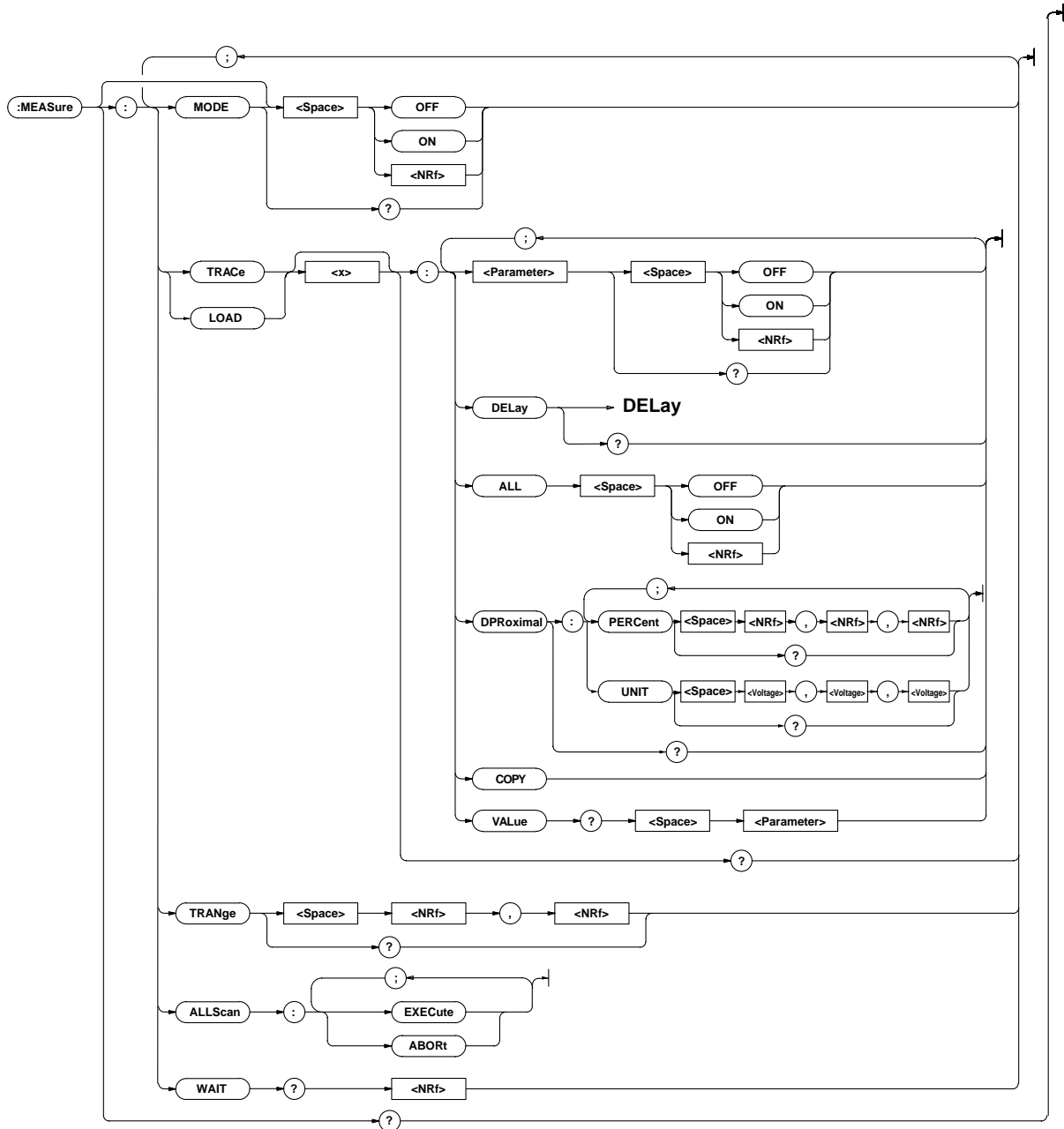
**Syntax** MATH:SENSitivity {<NRf>}  
MATH:SENSitivity?  
{<NRf>}=0.50 to 4.00 (div, 0.01 step)

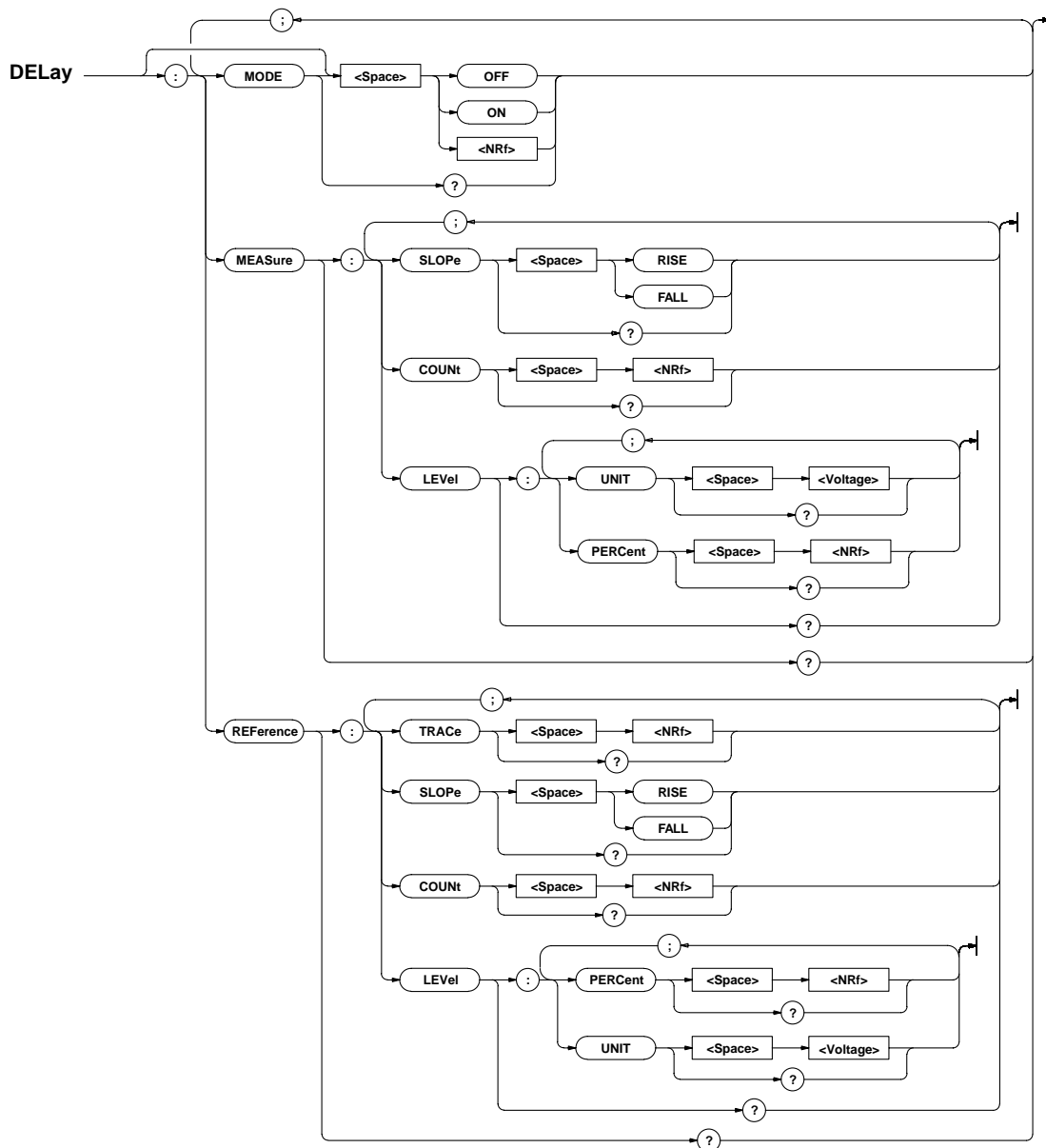
**Example** MATH:SENSITIVITY 0  
MATH:SENSITIVITY?→:MATH:  
SENSITIVITY 0.000



### 4.17 MEASure Group

The commands in the MEASure group are used to make settings relating to, and inquiries about automatic measurement of waveform parameters. You can make the same settings as when using the MEASURE key on the front panel.





**MEASure?**

**Function** Queries all the settings relating to automatic measurement of waveform parameters.

**Syntax** MEASure?

**Example** MEASURE?→:MEASURE:MODE 0;TRACE1:PTOPEAK 0;MAXIMUM 0;MINIMUM 0;RMS 0;AVERAGE 0;NOVERSHOOT 0;POVERSHOOT 0;HIGH 0;LOW 0;RISE 0;FALL 0;FREQUENCY 0;PERIOD 0;PWIDTH 0;NWIDTH 0;DUTYCYCLE 0;BWIDTH 0;TY1INTEG 0;TY2INTEG 0;PNUMBER 0;XY1INTEG 0;XY2INTEG 0;DPROXIMAL:PERCENT90,50,10;:MEASURE:TRACE1:DELAY:MODE 0;MEASURE:SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:TRACE1:DELAY:REFERENCE:TRACE 1;SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:TRACE2:PTOPEAK 0;MAXIMUM 0;MINIMUM 0;RMS 0;AVERAGE 0;NOVERSHOOT 0;POVERSHOOT 0;HIGH 0;LOW 0;RISE 0;FALL 0;FREQUENCY 0;PERIOD 0;PWIDTH 0;NWIDTH 0;DUTYCYCLE 0;BWIDTH 0;TY1INTEG 0;TY2INTEG 0;PNUMBER 0;XY1INTEG 0;XY2INTEG 0;DPROXIMAL:PERCENT 90,50,10;:MEASURE:TRACE2:DELAY:MODE 0;MEASURE:SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:TRACE2:DELAY:REFERENCE:TRACE 2;SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:TRACE3:PTOPEAK 0;MAXIMUM 0;MINIMUM 0;RMS 0;AVERAGE 0;NOVERSHOOT 0;POVERSHOOT 0;HIGH 0;LOW 0;RISE 0;FALL 0;FREQUENCY 0;PERIOD 0;PWIDTH 0;NWIDTH 0;DUTYCYCLE 0;BWIDTH 0;TY1INTEG 0;TY2INTEG 0;PNUMBER 0;XY1INTEG 0;XY2INTEG 0;DPROXIMAL:PERCENT 90,50,10;:MEASURE:TRACE3:DELAY:MODE 0;MEASURE:SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:TRACE3:DELAY:REFERENCE:TRACE 3;SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:TRACE4:PTOPEAK 0;MAXIMUM 0;MINIMUM 0;RMS 0;AVERAGE 0;NOVERSHOOT 0;POVERSHOOT 0;HIGH 0;LOW 0;RISE 0;FALL 0;FREQUENCY 0;PERIOD 0;PWIDTH 0;NWIDTH 0;DUTYCYCLE 0;BWIDTH 0;TY1INTEG 0;TY2INTEG 0;PNUMBER 0;XY1INTEG 0;XY2INTEG 0;DPROXIMAL:PERCENT 90,50,10;:MEASURE:TRACE4:DELAY:MODE 0;MEASURE:SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:TRACE4:DELAY:REFERENCE:TRACE 4;SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:LOAD1:PTOPEAK 0;MAXIMUM 0;MINIMUM 0;RMS 0;AVERAGE 0;NOVERSHOOT 0;POVERSHOOT 0;HIGH 0;LOW 0;RISE 0;FALL 0;FREQUENCY 0;PERIOD 0;PWIDTH 0;NWIDTH 0;DUTYCYCLE 0;BWIDTH 0;TY1INTEG 0;TY2INTEG 0;PNUMBER 0;XY1INTEG 0;XY2INTEG 0;DPROXIMAL:PERCENT 90,50,10;:MEASURE:LOAD1:DELAY:MODE 0;MEASURE:SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:LOAD1:DELAY:REFERENCE:TRACE 1;SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:LOAD2:

PTOPEAK 0;MAXIMUM 0;MINIMUM 0;RMS 0;AVERAGE 0;NOVERSHOOT 0;POVERSHOOT 0;HIGH 0;LOW 0;RISE 0;FALL 0;FREQUENCY 0;PERIOD 0;PWIDTH 0;NWIDTH 0;DUTYCYCLE 0;BWIDTH 0;TY1INTEG 0;TY2INTEG 0;PNUMBER 0;XY1INTEG 0;XY2INTEG 0;DPROXIMAL:PERCENT 90,50,10;:MEASURE:LOAD2:DELAY:MODE 0;MEASURE:SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:LOAD2:DELAY:REFERENCE:TRACE 1;SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:LOAD3:PTOPEAK 0;MAXIMUM 0;MINIMUM 0;RMS 0;AVERAGE 0;NOVERSHOOT 0;POVERSHOOT 0;HIGH 0;LOW 0;RISE 0;FALL 0;FREQUENCY 0;PERIOD 0;PWIDTH 0;NWIDTH 0;DUTYCYCLE 0;BWIDTH 0;TY1INTEG 0;TY2INTEG 0;PNUMBER 0;XY1INTEG 0;XY2INTEG 0;DPROXIMAL:PERCENT 90,50,10;:MEASURE:LOAD3:DELAY:MODE 0;MEASURE:SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:LOAD3:DELAY:REFERENCE:TRACE 1;SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:LOAD4:PTOPEAK 0;MAXIMUM 0;MINIMUM 0;RMS 0;AVERAGE 0;NOVERSHOOT 0;POVERSHOOT 0;HIGH 0;LOW 0;RISE 0;FALL 0;FREQUENCY 0;PERIOD 0;PWIDTH 0;NWIDTH 0;DUTYCYCLE 0;BWIDTH 0;TY1INTEG 0;TY2INTEG 0;PNUMBER 0;XY1INTEG 0;XY2INTEG 0;DPROXIMAL:PERCENT 90,50,10;:MEASURE:LOAD4:DELAY:MODE 0;MEASURE:SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:LOAD4:DELAY:REFERENCE:TRACE 1;SLOPE FALL;COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:TRANGE -5.00,5.00

**MEASure:ALLScan:ABORT**

**Function** Aborts automatic measurement of ACQ waveform data within cursor-designated measurement range.

**Syntax** MEASure:ALLScan:ABORT

**Example** MEASURE:ALLSCAN:ABORT

**MEASure:ALLScan:EXECute**

**Function** Starts automatic measurement of ACQ waveform data within cursor-designated measurement range. This is an overlapped command.

**Syntax** MEASure:ALLScan:EXECute

**Example** MEASURE:ALLSCAN:EXECUTE

**MEASure:{LOAD<x>|TRACe<x>}?**

**Function** Queries all the settings relating to the measurement items for each waveform.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}?  
<x>=1 to 4

**Example** MEASURE:TRACE1?→:MEASURE:TRACE1:  
PTOPEAK 0;MAXIMUM 0;MINIMUM 0;RMS 0;  
AVERAGE 0;NOVERSHOOT 0;POVERSHOOT 0;  
HIGH 0;LOW 0;RISE 0;FALL 0;FREQUENCY 0;  
PERIOD 0;PWIDTH 0;NWIDTH 0;DUTYCYCLE 0;  
BWIDTH 0;TY1INTEG 0;TY2INTEG 0;PNUMBER 0;  
XY1INTEG 0;XY2INTEG 0;DPROXIMAL:  
PERCENT 90,50,10;:MEASURE:TRACE1:DELAY:  
MODE 0;MEASURE:SLOPE FALL;COUNT 1;LEVEL:  
UNIT 0.0E+00;:MEASURE:TRACE1:DELAY:  
REFERENCE:TRACE 1;SLOPE FALL;COUNT 1;  
LEVEL:UNIT 0.0E+00

**MEASure:{LOAD<x>|TRACe<x>}:ALL**

**Function** Sets the parameters for the specified waveform ON/OFF.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:ALL  
{<Boolean>}  
<x>=1 to 4

**Example** (An Example for trace 1 is given)  
MEASURE:TRACE1:ALL OFF

**MEASure:{LOAD<x>|TRACe<x>}:COPY**

**Function** Copies the ON/OFF settings of all measurement items of a specific channel to all other waveforms.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:COPY  
**Example** MEASURE:TRACE1:COPY

**MEASure:{LOAD<x>|TRACe<x>}:DELay?**

**Function** Queries all the settings relating to delay measurement between channels for a specified waveform.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay?  
<x>=1 to 4

**Example** MEASURE:TRACE1:DELAY?→:MEASURE:TRACE1:  
DELAY:MODE 0;MEASURE:SLOPE RISE;  
COUNT 1;LEVEL:UNIT 0.0E+00;:MEASURE:  
TRACE1:DELAY:REFERENCE:TRACE 1;  
SLOPE RISE;COUNT 1;LEVEL:UNIT 0.0E+00

**MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure?**

**Function** Queries all the settings relating to the waveform which is the subject of delay measurement between channels for a specified waveform.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure?  
<x>=1 to 4

**Example** MEASURE:TRACE1:DELAY:MEASURE?→:MEASURE:  
TRACE1:DELAY:MEASURE:SLOPE RISE;  
COUNT 1;LEVEL:UNIT 0.0E+00

**MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:COUNT**

**Function** Sets the count number of the waveform which is the subject of delay measurement between channels for a specified waveform, or queries the current setting.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:  
COUNT {<Nrf>}  
MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:  
COUNT?  
<x>=1 to 4  
{<Nrf>}=1 to 9

**Example** MEASURE:TRACE1:DELAY:MEASURE:COUNT 2  
MEASURE:TRACE1:DELAY:MEASURE:COUNT?→:  
MEASURE:TRACE1:DELAY:MEASURE:COUNT 2

**MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:LEVEL?**

**Function** Queries all the settings relating to the threshold level of the waveform which is the subject of delay measurement between channels for a specified waveform.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:  
LEVEL?  
<x>=1 to 4

**Example** MEASURE:TRACE1:DELAY:MEASURE:LEVEL?→:  
MEASURE:TRACE1:DELAY:MEASURE:LEVEL:  
PERCENT 50

**Description** The result will be given in % or in the corresponding units, whichever is valid.

**MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:LEVEL:PERCent**

**Function** Sets the % value of the threshold level of the waveform which is the subject of delay measurement between channels for a specified waveform, or queries the current setting.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:  
LEVEL:PERCent {<Nrf>}  
MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:  
LEVEL:PERCent?  
<x>=1 to 4  
{<Nrf>}=0 to 100

**Example** MEASURE:TRACE1:DELAY:MEASURE:LEVEL:  
PERCENT 50  
MEASURE:TRACE1:DELAY:MEASURE:LEVEL:  
PERCENT?→:MEASURE:TRACE1:DELAY:MEASURE:LEVEL:  
PERCENT 50

**Description** If no % is set, this query will result in an error.

**MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:LEVel:UNIT**

**Function** Sets the voltage value of the treshold level of the waveform which is the subject of delay measurement between channels for a specified waveform, or queries the current setting.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:LEVel:UNIT <voltage>  
MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:LEVel:UNIT?  
<x>=1 to 4  
<voltage>=±8div (1/25 step of V/div)

**Example** MEASURE:TRACE1:DELAY:MEASURE:LEVEL:UNIT 50V  
MEASURE:TRACE1:DELAY:MEASURE:LEVEL:UNIT?→:MEASURE:TRACE1:DELAY:MEASURE:LEVEL:UNIT 50.0E+00

**Description** If no UNIT value is set, this query will result in an error.

**MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:SLOPe**

**Function** Sets the slope of the waveform which is the subject of delay measurement between channels for a specified waveform, or queries the current setting.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:SLOPe {RISE|FALL}  
MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:SLOPe?  
<x>=1 to 4

**Example** MEASURE:TRACE1:DELAY:MEASURE:SLOPE RISE  
MEASURE:TRACE1:DELAY:MEASURE:SLOPE?→:  
MEASURE:TRACE1:DELAY:MEASURE:SLOPE RISE

**MEASure:{LOAD<x>|TRACe<x>}:DELay[:MODE]**

**Function** Sets the delay measurement between channels for a specified waveform ON/OFF, or queries the current setting.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay[:MODE] {<Boolean>}  
MEASure:{LOAD<x>|TRACe<x>}:DELay:MODE?  
<x>=1 to 4

**Example** MEASURE:TRACE1:DELAY:MODE ON  
MEASURE:TRACE1:DELAY:MODE?→:MEASURE:TRACE1:DELAY:MODE 1

**MEASure:{LOAD<x>|TRACe<x>}:DELay:REFERENCE?**

**Function** Queries all settings relating to the reference waveform of delay measurement between channels for a specified waveform.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay:REFERENCE?  
<x>=1 to 4

**Example** MEASURE:TRACE1:DELAY:REFERENCE?→:  
MEASURE:TRACE1:DELAY:REFERENCE:TRACE 1;SLOPE RISE;COUNT 1;LEVEL:UNIT 0.0E+00

**MEASure:{LOAD<x>|TRACe<x>}:DELay:REFERENCE:COUNT**

**Function** Sets the count number of the reference waveform of delay measurement between channels for a specified channel, or queries the current setting.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay:REFERENCE:COUNT {<NRF>}  
MEASure:{LOAD<x>|TRACe<x>}:DELay:REFERENCE:COUNT?  
<x>=1 to 4  
{<NRF>}=1 to 9

**Example** MEASURE:TRACE1:DELAY:REFERENCE:COUNT 2  
MEASURE:TRACE1:DELAY:REFERENCE:COUNT?→:  
MEASURE:TRACE1:DELAY:REFERENCE:COUNT 2

**MEASure:{LOAD<x>|TRACe<x>}:DELay:REFERENCE:LEVEL?**

**Function** Queries all the settings relating to the treshold level of the reference waveform of delay measurement between channels for a specified waveform.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay:REFERENCE:LEVEL?  
<x>=1 to 4

**Example** MEASURE:TRACE1:DELAY:REFERENCE:LEVEL?→:  
MEASURE:TRACE1:DELAY:REFERENCE:LEVEL:PERCENT 50

**Description** The result will be given in % or in the corresponding units, whichever is valid.

**MEASure:{LOAD<x>|TRACe<x>}:DELay:REFERENCE:LEVEL:PERCENT**

**Function** Sets the % value of the treshold level of the reference waveform of delay measurement between channels for a specified waveform, or queries the current setting.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay:REFERENCE:LEVEL:PERCENT {<NRF>}  
MEASure:{LOAD<x>|TRACe<x>}:DELay:REFERENCE:LEVEL:PERCENT?  
<x>=1 to 4  
{<NRF>}=0 to 100

**Example** MEASURE:TRACE1:DELAY:REFERENCE:LEVEL:PERCENT 50  
MEASURE:TRACE1:DELAY:REFERENCE:LEVEL:PERCENT?→:MEASURE:TRACE1:DELAY:REFERENCE:LEVEL:PERCENT 50

**Description** If no % is set, this query will result in an error.

**MEASure:{LOAD<x>|TRACe<x>}:DELay:REFerence:LEVel:UNIT**

**Function** Sets the voltage value of the treshold level of the reference waveform of delay measurement between channels for a specified waveform, or queries the current setting.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay:  
REFerence:LEVel:UNIT <voltage>  
MEASure:{LOAD<x>|TRACe<x>}:DELay:  
REFerence:LEVel:UNIT?  
<x>=1 to 4  
<voltage>=±8div (1/25 step of V/div)

**Example** MEASURE:TRACE1:DELAY:REFERENCE:LEVEL:  
UNIT 50V  
MEASURE:TRACE1:DELAY:REFERENCE:LEVEL:  
UNIT?→:MEASURE:TRACE1:DELAY:REFERENCE:LEVEL:  
UNIT 50.0E+00

**Description** If no UNIT value is set, this query will result in an error.

**MEASure:{LOAD<x>|TRACe<x>}:DELay:REFerence:SLOPe**

**Function** Sets the slope of the reference waveform of delay measurement between channels for a specified waveform, or queries the current setting.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay:  
REFerence:SLOPe {RISE|FALL}  
MEASure:{LOAD<x>|TRACe<x>}:DELay:  
REFerence:SLOPe?  
<x>=1 to 4

**Example** MEASURE:TRACE1:DELAY:REFERENCE:SLOPE RISE  
MEASURE:TRACE1:DELAY:REFERENCE:SLOPE?→:  
MEASURE:TRACE1:DELAY:REFERENCE:SLOPE RISE

**MEASure:{LOAD<x>|TRACe<x>}:DELay:REFerence:TRACe**

**Function** Sets the trace of the reference waveform of delay measurement between channels for a specified waveform, or queries the current setting.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DELay:  
REFerence:TRACe {<NRf>}  
MEASure:{LOAD<x>|TRACe<x>}:DELay:  
REFerence:TRACe?  
<x>=1 to 4  
{<NRf>}=1 to 8 (5 to 8 correspond to  
Load1 to Load4)

**Example** MEASURE:TRACE1:DELAY:REFERENCE:TRACE 2  
MEASURE:TRACE1:DELAY:REFERENCE:TRACE?→:  
MEASURE:TRACE1:DELAY:REFERENCE:TRACE 2

**MEASure:{LOAD<x>|TRACe<x>}:DPRoximal?**

**Function** Queries all settings relating to distal, mesial, and proximal points.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DPRoximal?  
<x>=1 to 4

**Example** MEASURE:TRACE1:DPROXIMAL?  
→:MEASURE:TRACE1:DPROXIMAL:  
PERCENT 90,50,10

**MEASure:{LOAD<x>|TRACe<x>}:DPRoximal:PERCent**

**Function** Sets the distal, mesial, and proximal points in percentages, or queries the current settings.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DPRoximal:  
PERCent {<NRf>,<NRf>,<NRf>}  
MEASure:{LOAD<x>|TRACe<x>}:DPRoximal:  
PERCent?  
<x>=1 to 4  
<NRf>=0 to 100(in step of 1%)

**Example** MEASURE:TRACE1:DPROXIMAL:PERCENT 80,60,40  
MEASURE:TRACE1:DPROXIMAL:PERCENT?  
→:MEASURE:TRACE1:DPROXIMAL:  
PERCENT 80,60,40

**MEASure:{LOAD<x>|TRACe<x>}:DPRoximal:UNIT**

**Function** Sets the distal, mesial, and proximal points in terms of voltages, or queries the current settings.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:DPRoximal:UNIT  
{<voltage>,<voltage>,<voltage>}  
MEASure:{LOAD<x>|TRACe<x>}:DPRoximal:  
UNIT?  
<x>=1 to 4  
<voltage>=±8div (1/25 step of V/div)

**Example** MEASURE:TRACE1:DPROXIMAL:UNIT 50V,0V,-50V  
MEASURE:TRACE1:DPROXIMAL:UNIT?  
→:MEASURE:TRACE1:DPROXIMAL:  
UNIT 50.0E+00,0.0E+00,-50.0E+00

**MEASure[:MODE]**

**Function** Turns automated measurement ON/OFF, or queries the current setting.

**Syntax** MEASure[:MODE] {<Boolean>}  
MEASure:MODE?

**Example** MEASURE:MODE OFF  
MEASURE:MODE?→:MEASURE:MODE 0

**MEASure:{LOAD<x>|TRACe<x>}:<parameter>**

**Function** Turns a measurement item ON/OFF for a specified waveform, or queries the current setting.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:  
<parameter> {<Boolean>}  
MEASure:{LOAD<x>|TRACe<x>}:<parameter>?  
<x>=1 to 4  
<parameter>= {AVERage|BWIth|DUTYcycLe|  
FALL|FREQuency|HIGH|LOW|  
MAXimum|MINimum|NOVershoot|  
NWIth|PERiod|POVershoot|  
PNUMber|PTOPeak|PWIDth|RISE|  
RMS|TY1Integ|TY2Integ|XY1Integ|  
XY2Integ}

**Example** (An example for averaging on trace 1 is given)  
MEASURE:TRACE1:AVERAGE OFF  
MEASURE:TRACE1:AVERAGE?→:MEASURE:TRACE1:  
AVERAGE 0

**MEASure:TRANge**

**Function** Sets the measurement range, or queries the current setting.

**Syntax** MEASure:TRANge {<NRf>},{<NRf>}  
 MEASure:TRANge?  
 {<NRf>}=-5.00 to 5.00 (in steps of  
 0.02div)

**Example** MEASURE:TRANGE 2.50,-2.50  
 MEASURE:TRANGE?→:MEASURE:TRANGE -2.50,  
 2.50

**MEASure:{LOAD<x>|TRACe<x>}:VALue?****<parameter>**

**Function** Queries the automatically measured value of the specified measurement item for the specified waveform.

**Syntax** MEASure:{LOAD<x>|TRACe<x>}:VALue?  
 <parameter>  
 <x>=1 to 4  
 <parameter>= {AVERAge|BWIDth|DUTYcycle|  
 DELay|FALL|FREQuency|HIGH|LOW|  
 MAXimum|MINimum|NOVershoot|  
 NWIDth|PERiod|POVershoot|  
 PNUMber|PTOPeak|PWIDth|RISE|  
 RMS|TY1Integ|TY2Integ|  
 XY1Inte g|XY2Integ}

**Example** (An example for averaging on trace 1 is given)  
 MEASURE:TRACE1:VALUE?  
 AVERAGE→:MEASURE:TRACE1:VALUE 4.95E-03

**Description** In case the mode is OFF, or measurement is not possible for any other reason, "NAN" (non-numeric) will appear.

**MEASure:WAIT?**

**Function** Wait for automatic measurement to finish with time-out option.

**Syntax** MEASure:WAIT? {<NRf>}  
 {<NRf>}=1 to 36000 (time-out period, in  
 units of 100 ms)

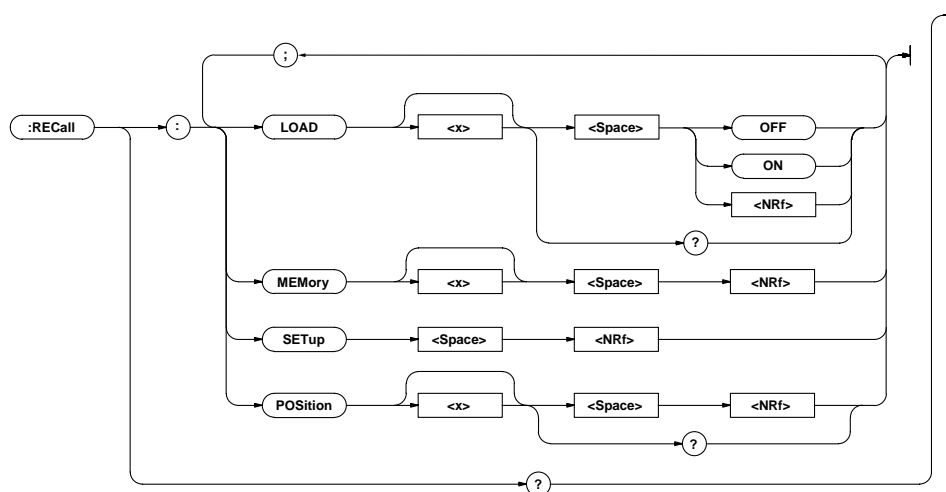
**Example** MEASURE:WAIT?→:MEASURE:WAIT 0

**Discription**

- If automatic measurement finishes before the time-out, "0" is returned. If the measurement has not finished or if automatic measurement is not being made, "1" is returned.
- "0" is returned at the time when the automatic measurement finishes, regardless of the time set for the time-out.
- This command can not be used to wait for the automatic measurement of the ACQ waveform data to finish.

## 4.18 RECall Group

The RECall group is used to recall data from the internal memory. You can use it to perform the same operations as when using the STORE/RECALL key on the front panel.



### RECall?

**Function** Queries all settings relating to the recalling of data from the internal memory.

**Syntax** RECall?

**Example** RECALL?→:RECALL:LOAD1 0;LOAD2 0;LOAD3 0;  
LOAD4 0;POSITION1 0.000;POSITION2 0.000;  
POSITION3 0.000;POSITION4 0.000

### RECall:LOAD<x>

**Function** Sets the display of the loaded waveform ON/OFF, or queries the current setting.

**Syntax** RECall:LOAD<x> {<Boolean>}

RECall:LOAD?

<x>=1 to 4

**Example** RECALL:LOAD1 ON  
RECALL:LOAD1?→:RECALL:LOAD1 1

### RECall:MEMory<x>

**Function** Recalls waveform data from the specified internal memory and recalls it as a specified trace.

**Syntax** RECall:MEMory<x> {<NRf>}

<x>=1 to 4

{<NRf>}=1 to 4

**Example** (An example in which waveform data from memory 1 is recalled as trace 2 is given below)  
RECALL:MEMORY1 2

### RECall:POSITION<x>

**Function** Sets the vertical position of the recalled waveform, or queries the current setting.

**Syntax** RECall:POSITION<x> {<NRf>}

RECall:POSITION<x>?

<x>=1 to 4

{<NRf>}=-4.00 to 4.00(in steps of  
0.02div)

**Example** RECALL:POSITION1 0  
RECALL:POSITION1?→:RECALL:POSITION 0.00

### RECall:SETup

**Function** Recalls the specified set-up data from the internal memory.

**Syntax** RECall:SETup {<NRf>}

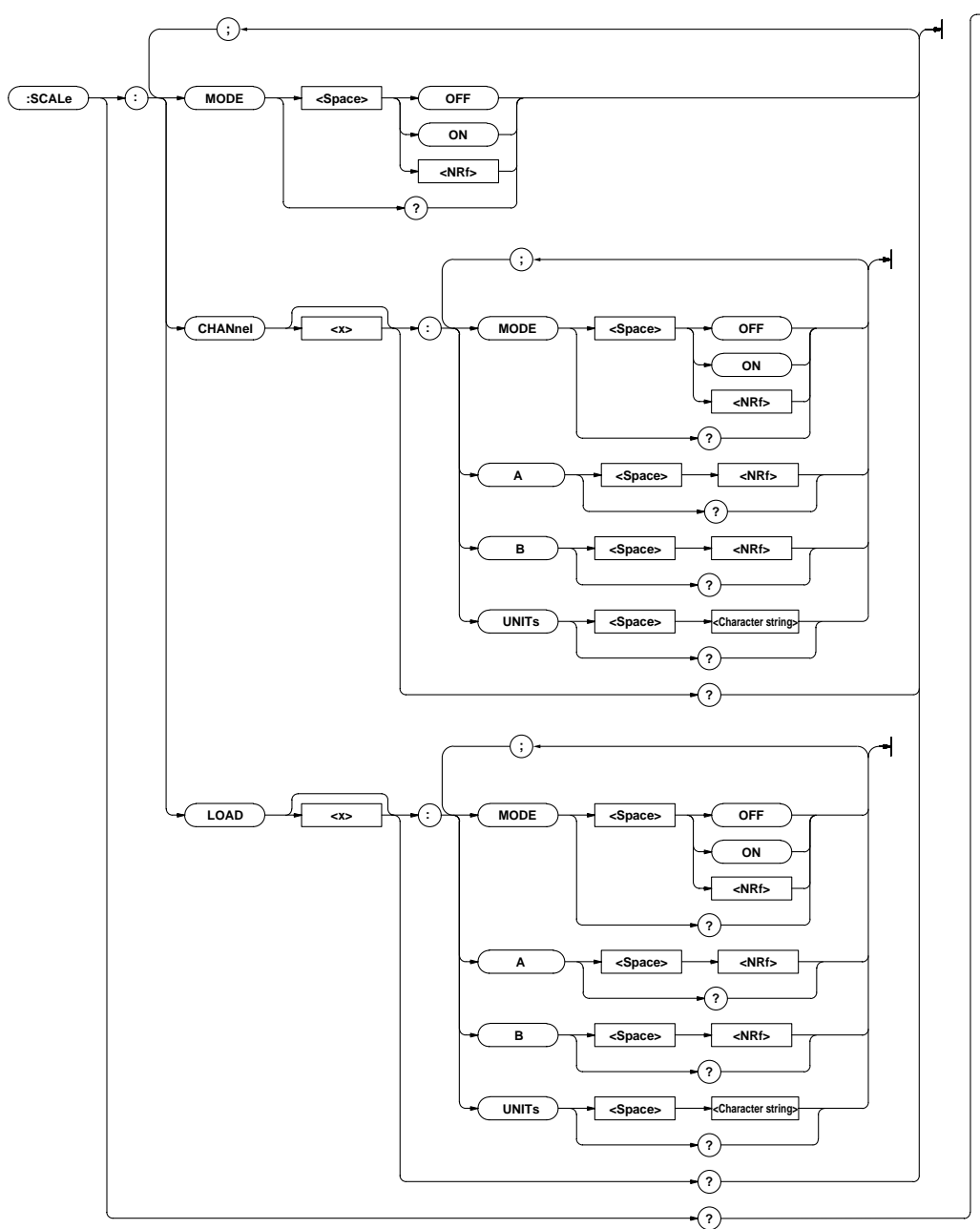
{<NRf>}=1 to 4

**Example** RECALL:SETUP 1



## 4.19 SCALE Group

The SCALE group is used for executing linear scaling. You can use it to perform the same operations as when using the MISC key on the front panel followed by the settings on the Scale menu.



### SCALE?

**Function** Queries all the settings relating to linear scaling.

**Syntax** SCALE?

**Example** SCALE?→:SCALE:MODE 0;CHANNEL1:MODE 0;  
A 0.0E+00;B 0.0E+00;UNIT “”;:SCALE:  
CHANNEL2:MODE 0;A 0.0E+00;B 0.0E+00;  
UNIT “”;:SCALE:CHANNEL3:MODE 0;A 0.0E+00;  
B 0.0E+00;UNIT “”;:SCALE:CHANNEL4:MODE 0;  
A 0.0E+00;B 0.0E+00;UNIT “”;:SCALE:LOAD1:  
MODE 0;A 0.0E+00;B 0.0E+00;UNIT “”;:  
SCALE:LOAD2:MODE 0;A 0.0E+00;B 0.0E+00;  
UNIT “”;:SCALE:LOAD3:MODE 0;A 0.0E+00;  
B 0.0E+00;UNIT “”;:SCALE:LOAD4:MODE 0;  
A 0.0E+00;B 0.0E+00;UNIT “”

### SCALE:CHANnel<x>?

**Function** Queries all the settings relating to linear scaling for a specified channel.

**Syntax** SCALE:CHANnel<x>?  
<x>=1 to 4

**Example** SCALE:CHANNEL1?→:SCALE:CHANNEL1:MODE 0;  
A 0.0E+00;B 0.0E+00;UNIT “”

**SCALE:CHANnel<x>:A**

**Function** Sets the linear scaling A value for a specified channel, or queries the current setting.

**Syntax** SCALE:CHANnel<x>:A {<NRf>}  
SCALE:CHANnel<x>:A?  
<x>=1 to 4

{<NRf>}=-3.4E+38 to 3.4E+38  
**Example** SCALE:CHANNEL1:A 1.5  
SCALE:CHANNEL1:A?→:SCALE:CHANNEL1:  
A 1.500000E+00

**SCALE:CHANnel<x>:B**

**Function** Sets the linear scaling B value for a specified channel, or queries the current setting.

**Syntax** SCALE:CHANnel<x>:B {<NRf>}  
SCALE:CHANnel<x>:B?  
<x>=1 to 4

{<NRf>}=-3.4E+38 to 3.4E+38  
**Example** SCALE:CHANNEL1:B 1.5  
SCALE:CHANNEL1:B?→:SCALE:CHANNEL1:  
B 1.500000E+00

**SCALE:CHANnel<x>:MODE**

**Function** Sets linear scaling for a specific channel ON/OFF, or queries the current setting.

**Syntax** SCALE:CHANnel<x>:MODE {<Boolean>}  
SCALE:CHANnel<x>:MODE?  
<x>=1 to 4

**Example** SCALE:CHANNEL1:MODE ON  
SCALE:CHANNEL1:MODE?→:SCALE:CHANNEL1:  
MODE 1

**SCALE:CHANnel<x>:UNIT**

**Function** Sets the linear scaling unit for a specific channel, or queries the current setting.

**Syntax** SCALE:CHANnel<x>:UNIT <character>  
SCALE:CHANnel<x>:UNIT?  
<x>=1 to 4

<character>= Up to 8 characters, refer to User's Manual IM701530-01E.  
**Example** SCALE:CHANNEL1:UNIT "ABC"  
SCALE:CHANNEL1:UNIT?→:SCALE:CHANNEL1:  
UNIT "ABC"

**Description** When you set more than 8 characters, only the first 8 ones will be valid.

**SCALE:LOAD<x>?**

**Function** Queries all the settings relating to linear scaling for a specified loaded waveform.

**Syntax** SCALE:LOAD<x>?  
<x>=1 to 4

**Example** SCALE:LOAD1?→:SCALE:LOAD1:MODE 0;  
A 0.0E+00;B 0.0E+00;UNIT ""

**SCALE:LOAD<x>:A**

**Function** Sets the linear scaling A value for a specified loaded waveform, or queries the current setting.

**Syntax** SCALE:LOAD<x>:A {<NRf>}  
SCALE:LOAD<x>:A?  
<x>=1 to 4

{<NRf>}=-3.4E+38 to 3.4E+38  
**Example** SCALE:LOAD1:A 1.5  
SCALE:LOAD1:A?→:SCALE:LOAD1:  
A 1.500000E+00

**SCALE:LOAD<x>:B**

**Function** Sets the linear scaling B value for a specified loaded waveform, or queries the current setting.

**Syntax** SCALE:LOAD<x>:B {<NRf>}  
SCALE:LOAD<x>:B?  
<x>=1 to 4

{<NRf>}=-3.4E+38 to 3.4E+38  
**Example** SCALE:LOAD1:B 1.5  
SCALE:LOAD1:B?→:SCALE:LOAD1:  
B 1.500000E+00

**SCALE:LOAD<x>:MODE**

**Function** Sets linear scaling for a specified loaded waveform ON/OFF, or queries the current setting.

**Syntax** SCALE:LOAD<x>:MODE {<Boolean>}  
SCALE:LOAD<x>:MODE?  
<x>=1 to 4

**Example** SCALE:LOAD1:MODE ON  
SCALE:LOAD1:MODE?→:SCALE:LOAD1:MODE 1

**SCALE:LOAD<x>:UNIT**

**Function** Sets the linear scaling unit for a specified loaded waveform, or queries the current setting.

**Syntax** SCALE:LOAD<x>:UNIT <character>  
SCALE:LOAD<x>:UNIT?  
<x>=1 to 4

<character>= Up to 8 characters, refer to User's Manual IM701530-01E.  
**Example** SCALE:LOAD1:UNIT "ABC"  
SCALE:LOAD1:UNIT?→:SCALE:LOAD1:  
UNIT "ABC"

**Description** When you set more than 8 characters, only the first 8 ones will be valid.

**SCALE:MODE**

**Function** Sets linear scaling ON/OFF, or queries the current setting.

**Syntax** SCALE:MODE {<Boolean>}  
SCALE:MODE?

**Example** SCALE:MODE ON  
SCALE:MODE?→:SCALE:MODE 1

### 4.20 SNAP Group

The SNAP command is to execute a snapshot. The same function can be performed using the SNAP SHOT key on the front panel.



#### SNAP

- Function** Executes a snapshot.
- Syntax** SNAP
- Example** SNAP

### 4.21 START Group

The START command is used to start acquisition. The same function can be performed by pressing the START/STOP key on the front panel.

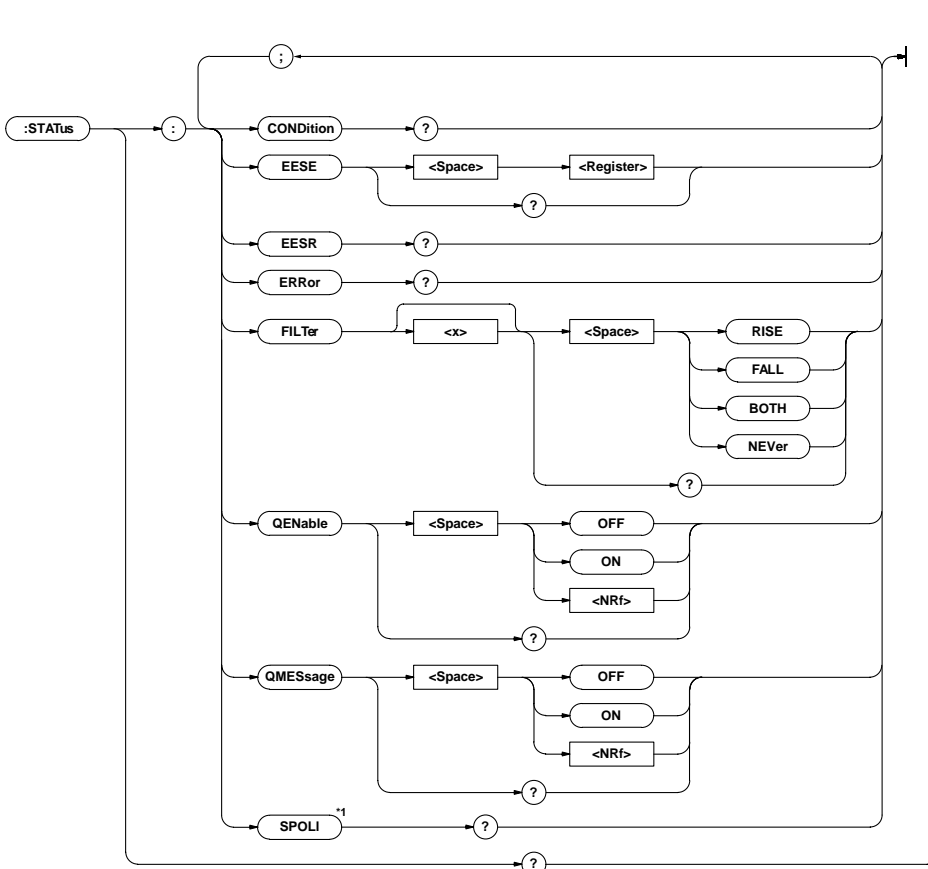


#### START

- Function** Starts waveform acquisition.
- Syntax** START
- Example** START

### 4.22 STATUS Group

The commands in the STATUS group are used to make settings relating to, and inquiries about the communications status function. There is no key on the front panel for this function. For details on the status report, refer to chapter 5.



\*1 Available to use the separately sold RS-232-C interface unit 700927.

**STATus?**

**Function** Queries all the settings relating to the status of the communication Function.

**Syntax** STATus?

**Example** STATus?→:STATUS:EESE 0;FILTER1 NEVER;  
FILTER2 NEVER;FILTER3 NEVER;  
FILTER4 NEVER;FILTER5 NEVER;  
FILTER6 NEVER;FILTER7 NEVER;  
FILTER8 NEVER;FILTER9 NEVER;  
FILTER10 NEVER;FILTER11 NEVER;  
FILTER12 NEVER;FILTER13 NEVER;  
FILTER14 NEVER;FILTER15 NEVER;  
FILTER16 NEVER;QMESSAGE 1

**STATus:CONDition?**

**Function** Queries the contents of the condition register.

**Syntax** STATus:CONDition?

**Example** STATus:CONDITION→16

**Description** For a Description of the condition register, refer to chapter 5.

**STATus:EESE**

**Function** Sets the extended event enable register, or queries the current setting.

**Syntax** STATus:EESE <Register>  
STATus:EESE?

<Register>=0 to 65535

**Example** STATus:EESE 257  
STATus:EESE?→:STATUS:EESE 257

**Description** For a Description of the extended event enable register, refer to chapter 5.

**STATus:EESR?**

**Function** Queries the contents of the extended event register and clears the register.

**Syntax** STATus:EESR?

**Example** STATus:EESR?→1

**Description** For a Description of the event register, refer to chapter 5.

**STATus:ERRor?**

**Function** Queries the code and the message (at the beginning of the error queue) of the error which has occurred.

**Syntax** STATus:ERRor?

**Example** STATus:ERROR?→113,"Undefine header"

**STATus:FILTer<x>**

**Function** Queries all the settings relating to the specified transition filter, or queries the current setting.

**Syntax** STATus:FILTer<x> {RISE|FALL|BOTH|NEVer}  
STATus:FILTer<x>?  
<x>=1 to 16

**Example** STATus:FILTER2 RISE  
STATus:FILTER2?→:STATUS:FILTER2 RISE

**Description** For a Description of the transition filter, refer to chapter 5.

**STATus:QENable**

**Function** Selects whether or not to store a message other than an error message into the error queue, or queries the current setting.

**Syntax** STATus:QENable {<Boolean>}  
STATus:QENable?

**Example** STATus:QENABLE ON  
STATus:QENABLE?→:STATUS:QENABLE 1

**STATus:QMESsage**

**Function** Selects whether or not to add the message contents to a response to "STATus:ERRor?", or queries the current setting.

**Syntax** STATus:QMESsage {<Boolean>}  
STATus:QMESsage?

**Example** STATus:QMESSAGE OFF  
STATus:QMESSAGE?→:STATUS:QMESSAGE 0

**STATus:SPOLL?(Serial Poll)**

**Function** Executes a serial poll.

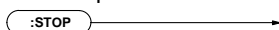
**Syntax** STATus:SPOLL?

**Example** STATus:SPOLL?→:STATUS:SPOLL 0

**Description** This command is available to use the separately sold RS-232-C interface unit 700927.

**4.23 STOP Group**

The STOP command is used to stop acquisition. The same function can be performed using the START/STOP key on the front panel.

**STOP**

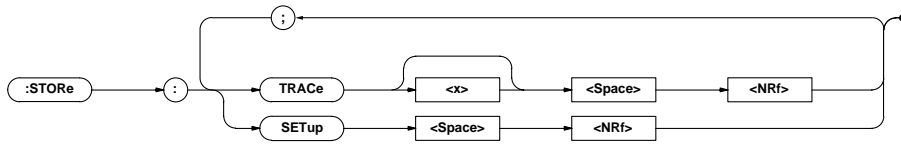
**Function** Stops waveform acquisition.

**Syntax** STOP

**Example** STOP

### 4.24 STORE Group

The commands in the STORE group are used to store data in the internal memory. The same function can be performed using the STORE/RECALL key on the front panel.



#### STORE:SETUp

**Function** Stores setting parameters in the internal memory.

**Syntax** STORE:SETUp {<NRf>}  
{<NRf>}=1 to 4

**Example** STORE:SETUP 1

#### STORE:TRACe<x>

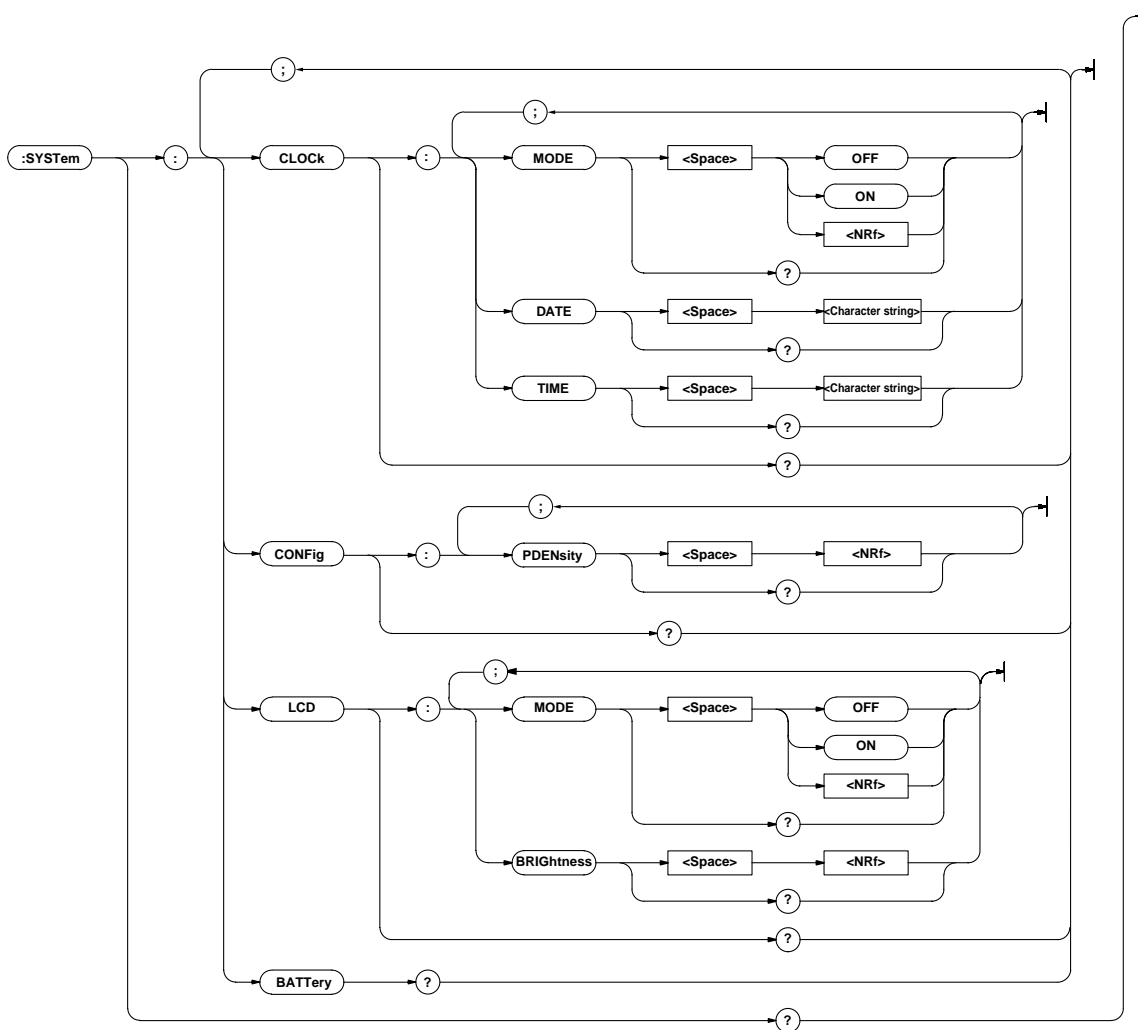
**Function** Stores waveform data in the internal memory.

**Syntax** STORE:TRACe<x> {<NRf>}  
<x>=1 to 4  
{<NRf>}=1 or 4

**Example** (An example of storing waveform data (trace 1) in memory No. 2 is given below.)  
STORE:TRACE1 2

### 4.25 SYSTEM Group

The commands in the SYSTEM group are used to make settings relating to, and inquiries about the system. The same settings and queries can be made using the MISC key on the front panel and selecting the Date Time, or Config menus.



**SYSTem?**

**Function** Queries all the system settings.  
**Syntax** SYSTem?  
**Example** SYSTem?→:SYSTem:CLOCK:MODE 1;  
 DATE "95/04/20";TIME "11:25:30";:SYSTem:  
 CONFIG:PDENSITY 1;:SYSTem:LCD:MODE 1;  
 BRIGHTNESS 1

**SYSTem:BATTery?**

**Function** Queries the condition of the internal lithium battery.  
**Syntax** SYSTem:BATTery?  
**Example** SYSTem:BATTery?→:SYSTem:BATTery 1  
**Description** "1" is returned if the battery is Functioning, "0" is returned if the battery has run out.

**SYSTem:CLOCK?**

**Function** Queries all the settings relating to the date and time.  
**Syntax** SYSTem:CLOCK?  
**Example** SYSTem:CLOCK?→:SYSTem:CLOCK:MODE 1;  
 DATE "93/11/30";TIME "14:30:00"

**SYSTem:CLOCK:DATE**

**Function** Sets the date, or queries the current setting.  
**Syntax** SYSTem:CLOCK:DATE <character string>  
 SYSTem:CLOCK:DATE?  
 <character string>= YY/MM/DD, refer to  
 User's Manual IM701530-01E.  
**Example** SYSTem:CLOCK:DATE "93/09/30"  
 SYSTem:CLOCK:DATE?→:SYSTem:CLOCK:  
 DATE "93/09/30"

**SYSTem:CLOCK:MODE**

**Function** Turns display of the date and time ON/OFF, or queries the current setting.  
**Syntax** SYSTem:CLOCK:MODE {<Boolean>}  
 SYSTem:CLOCK:MODE?  
**Example** SYSTem:CLOCK:MODE ON  
 SYSTem:CLOCK:MODE?→:SYSTem:CLOCK:MODE 1  
**Description** The "year" is represented with the lower two digits. Years 2000 to 2079 are represented by 00 to 79, and years 1980 to 1999 are represented by 80 to 99.

**SYSTem:CLOCK:TIME**

**Function** Sets the time, or queries the current setting.  
**Syntax** SYSTem:CLOCK:TIME <character string>  
 SYSTem:CLOCK:TIME?  
 <character string>= HH:MM:SS, refer to  
 User's Manual IM701530-01E.  
**Example** SYSTem:CLOCK:TIME "14:30:00"  
 SYSTem:CLOCK:TIME?→:SYSTem:CLOCK:  
 TIME "14:30:00"

**SYSTem:CONFIg?**

**Function** Queries all the settings relating to the configuration system.  
**Syntax** SYSTem:CONFIg?  
**Example** SYSTem:CONFIg?→:SYSTem:CONFIg:PDENSITY 2

**SYSTem:CONFIg:PDENsity**

**Function** Selects the level of the printing density, or queries the current setting.  
**Syntax** SYSTem:CONFIg:PDENsity {<Nrf>}  
 SYSTem:CONFIg:PDENsity?  
 {<Nrf>}=0 to 3  
**Example** SYSTem:CONFIg:PDENSITY 2  
 SYSTem:CONFIg:PDENSITY?→:SYSTem:CONFIg:  
 PDENSITY 2

**SYSTem:LCD?**

**Function** Queries all settings relating to the LCD.  
**Syntax** SYSTem:LCD?  
**Example** SYSTem:LCD?→:SYSTem:LCD:MODE 1;  
 BRIGHTNESS 1

**SYSTem:LCD:BRIGHtness**

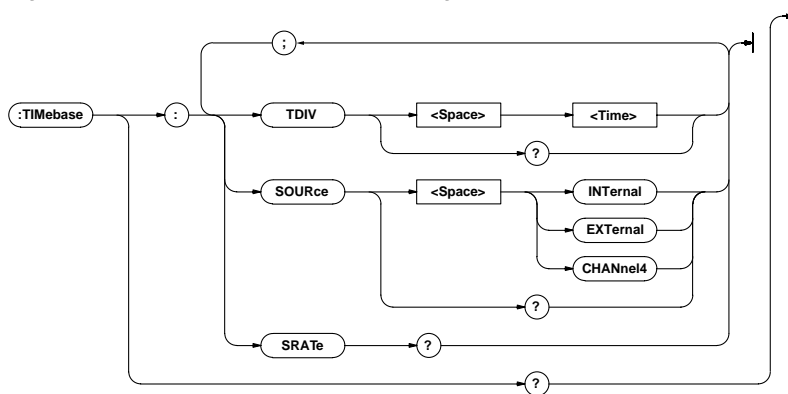
**Function** Sets the brightness of the LCD, or queries the setting.  
**Syntax** SYSTem:LCD:BRIGHtness {<Nrf>}  
 SYSTem:LCD:BRIGHtness?  
 <Nrf>=1 to 4  
**Example** SYSTem:LCD:BRIGHtnESS 2  
 SYSTem:LCD:BRIGHtnESS?→:SYSTem:LCD:  
 BRIGHtnESS 2

**SYSTem:LCD:MODE**

**Function** Turns the back light of the LCD ON/OFF, or queries the current setting.  
**Syntax** SYSTem:LCD:MODE {<Boolean>}  
 SYSTem:LCD:MODE?  
**Example** SYSTem:LCD:MODE OFF  
 SYSTem:LCD:MODE?→:SYSTem:LCD:MODE 0

## 4.26 TIMEbase Group

The commands in the TIMEbase group are used to make settings relating to, and inquiries about the timebase. These settings and inquiries can also be made using the TIME/DIV knob on the front panel.



### TIMEbase?

**Function** Queries all the timebase settings.

**Syntax** TIMEbase?

**Example** TIMEBASE?→:TIMEBASE:TDIV 1.000000E-03;  
SOURCE INTERNAL

### TIMEbase:SOURce

**Function** Sets the timebase, or queries the current setting.

**Syntax** TIMEbase:SOURce  
{CHANnel4|EXTernal|INTernal}  
TIMEbase:SOURce?

**Example** TIMEBASE:SOURCE INTERNAL  
TIMEBASE:SOURCE?→:TIMEBASE:  
SOURCE INTERNAL

### TIMEbase:SRATe?

**Function** Queries the current sample rate.

**Syntax** TIMEbase:SRATe?

**Example** TIMEBASE:SRATE?→:TIMEBASE:  
SRATE 1.000000E+06

**Description** The same inquiry can be made using  
"WAVEform:SRATE?".

### TIMEbase:TDIV

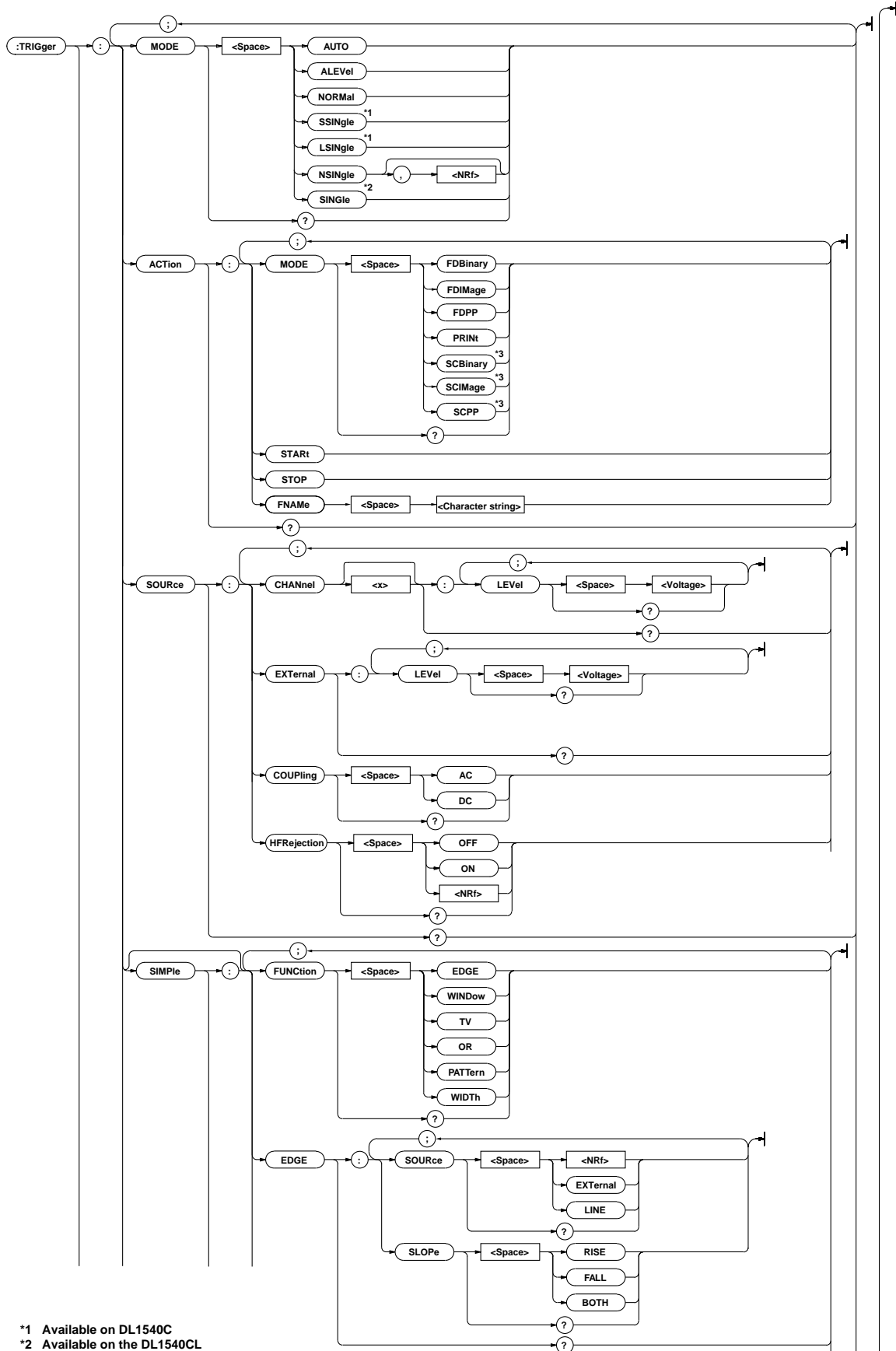
**Function** Set the Time axis, or queries the current setting.

**Syntax** TIMEbase:TDIV <Time>  
TIMEbase:TDIV?  
<Time>=2ns to 50s(in steps of 1-2-5)

**Example** TIMEBASE:TDIV 1MS  
TIMEBASE:TDIV?→:TIMEBASE:  
TDIV 1.000000E-03

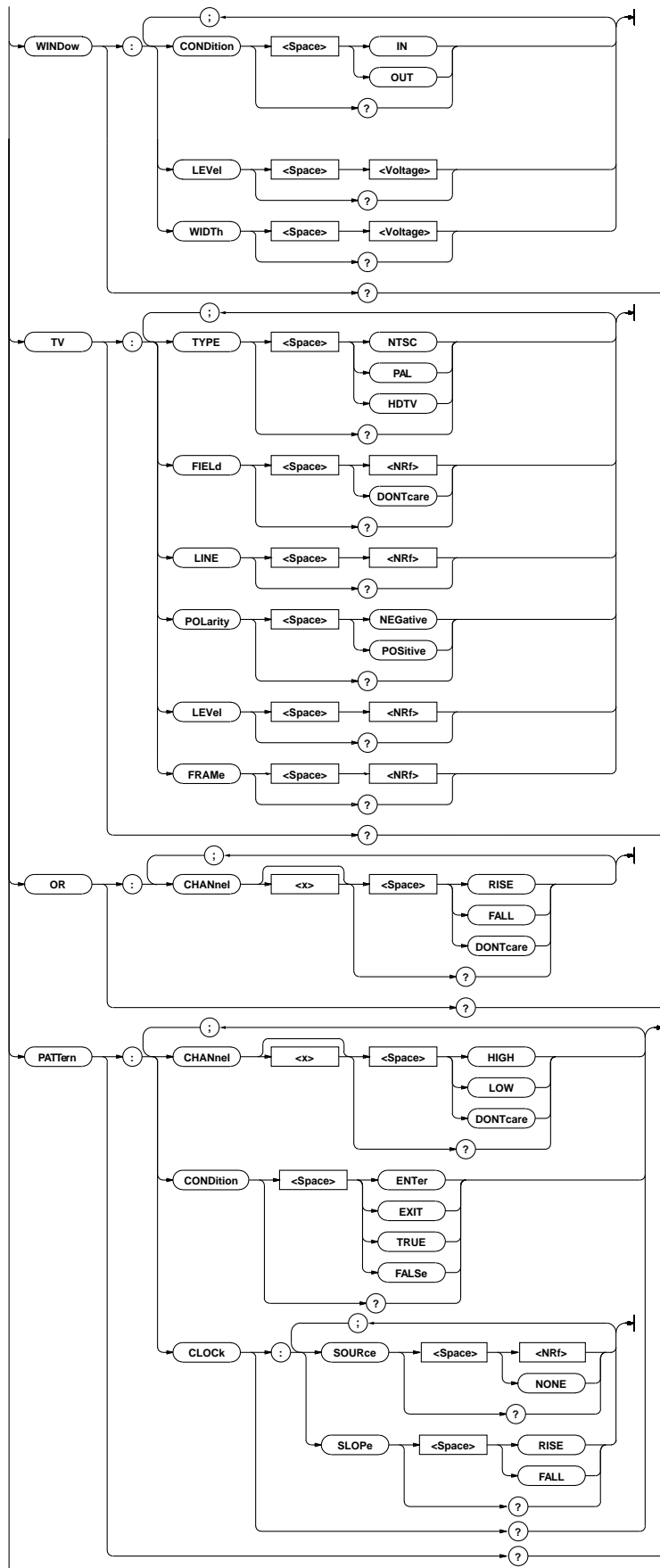
## 4.27 TRIGger Group

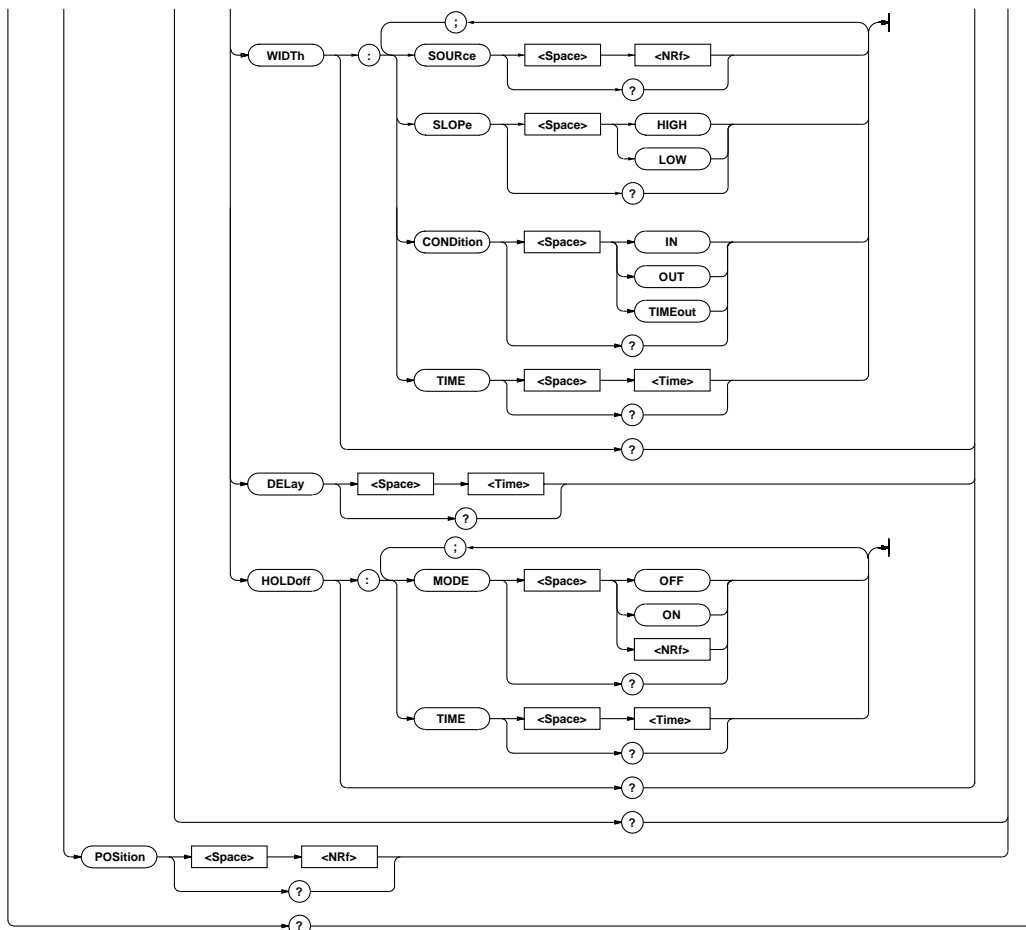
The commands in the TRIGger group are used to make settings relating to, and inquiries about triggers. The same settings and inquiries can be made using the TRIGGER group keys (the MODE key, the FUNCTION key, the POSITION key, LEVEL/SOURCE key, DELAY key, COUPLING key and HOLD OFF key). Note that the pattern trigger, OR trigger and pulse trigger are optional.



\*1 Available on DL1540C  
 \*2 Available on the DL1540CL  
 \*3 Available to use the separately sold SCSI interface unit 700930.







### TRIGger?

**Function** Queries all trigger settings.

**Syntax** TRIGger?

**Example** TRIGGER?→:TRIGGER:MODE AUTO;:TRIGGER:POSITION 0.00;SIMPLE:FUNCTION EDGE;EDGE:SOURCE 1;SLOPE RISE;:TRIGGER:SIMPLE:WINDOW:CONDITION IN;LEVEL 0.0E+00;WIDTH 50.0E+00;:TRIGGER:SIMPLE:TV:TYPE NTSC;FIELD 1;LINE 5;POLARITY NEGATIVE;LEVEL 0.50;FRAME 1;:TRIGGER:SIMPLE:OR:CHANNEL1 DONTCARE;CHANNEL2 DONTCARE;CHANNEL3 DONTCARE;CHANNEL4 DONTCARE;:TRIGGER:SIMPLE:PATTERN:CHANNEL1 DONTCARE;CHANNEL2 DONTCARE;CHANNEL3 DONTCARE;CHANNEL4 DONTCARE;CLOCK:SOURCE NONE;:TRIGGER:SIMPLE:PATTERN:CONDITION ENTER;:TRIGGER:SIMPLE:WIDTH:CONDITION IN;TIME 5.0E-09;SOURCE 1;SLOPE HIGH;:TRIGGER:SIMPLE:DELAY 0.0E+00;HOLDOFF:MODE 0;TIME 0.0E+00;:TRIGGER:SOURCE:COUPLING AC;HFREJECTION 0;CHANNEL1:LEVEL 0.0E+00;:TRIGGER:SOURCE:CHANNEL2:LEVEL 0.0E+00;:TRIGGER:SOURCE:CHANNEL3:LEVEL 0.0E+00;:TRIGGER:SOURCE:CHANNEL4:LEVEL 0.0E+00;:TRIGGER:SOURCE:EXTERNAL:LEVEL 150.0E-03;:TRIGGER:ACTION:MODE PRINT

### TRIGger:ACTion?

**Function** Queries all the settings relating to action-on-trigger.

**Syntax** TRIGger:ACTion?

**Example** TRIGGER:ACTION?→:TRIGGER:ACTION:MODE NONE

### TRIGger:ACTion:FNAME

**Function** Sets the file name when saving upon an action-on-trigger.

**Syntax** TRIGger:ACTion:FNAME <character string>  
<character string>= Up to 4 characters can be used, refer to User's manual IM701530-01E.

**Example** TRIGGER:ACTION:FNAME "TRIG"

**Description** This file name will be equipped with an automatically applied number.

### TRIGger:ACTion:MODE

**Function** Selects the action-on-trigger mode, or queries the current setting.

**Syntax** TRIGger:ACTion:MODE {FDBinary|FDPP|FDImage|PRINT|SCBinary|SCImage|SCPP}  
TRIGger:ACTion:MODE?

**Example** TRIGGER:ACTION:MODE FDPP  
TRIGGER:ACTION:MODE?→:TRIGGER:ACTION:MODE FDPP

**Description** SCPP, SCBinary and SCImage are available to use the separately sold SCSI interface unit 700930.

**TRIGger:ACTion:START**

**Function** Starts the action-on-trigger.

**Syntax** TRIGger:ACTion:START

**Example** TRIGGER:ACTION:START

**Description** In order to halt the trigger action, use "TRIGger:ACTION:STOP".

**TRIGger:ACTion:STOP**

**Function** Stops the action-on-trigger.

**Syntax** TRIGger:ACTion:STOP

**Example** TRIGGER:ACTION:STOP

**Description** This command only results in stopping the trigger action; the settings will not be canceled. "TRIGger:ACTION:START" will start the trigger action. In order to cancel the settings, set the "TRIGger:ACTION:MODE" to "NONE".

**TRIGger:MODE**

**Function** Sets the trigger mode, or queries the current setting.

**Syntax** TRIGger:MODE {ALEVEl|AUTO|NORMAl|NSINGle [, {<NRf>}]|LSINGle|SSINGle} (for DL1540C)  
TRIGger:MODE {ALEVEl|AUTO|NORMAl|NSINGle [, {<NRf>}]|SINGle}(for DL1540CL)  
TRIGger:MODE?

**Example** TRIGGER:MODE AUTO  
TRIGGER:MODE?→:TRIGGER:MODE AUTO

**TRIGger:POSition**

**Function** Sets the trigger position, or queries the current setting.

**Syntax** TRIGger:POSition {<NRf>}  
TRIGger:POSition?  
{<NRf>}=-5.00 to 5.00 (in 0.01div steps)

**Example** TRIGGER:POSITION 0.00  
TRIGGER:POSITION?→:TRIGGER:POSITION 0.00

**TRIGger:SIMple?**

**Function** Queries all trigger settings.

**Syntax** TRIGger:SIMple?

**Example** TRIGGER:SIMPLE?→:TRIGGER:SIMPLE:  
FUNCTION EDGE;EDGE:SOURCE 1;SLOPE RISE;;  
TRIGGER:SIMPLE:WINDOW:CONDITION IN;  
LEVEL 0.0E+00;WIDTH 50.0E+00;;TRIGGER:  
SIMPLE:TV:TYPE NTSC;FIELD 1;LINE 5;  
POLARITY NEGATIVE;LEVEL 0.50;FRAME 1;;  
TRIGGER:SIMPLE:OR:CHANNEL1 DONTCARE;  
CHANNEL2 DONTCARE;CHANNEL3 DONTCARE;  
CHANNEL4 DONTCARE;;TRIGGER:SIMPLE:  
PATTERN:CHANNEL1 DONTCARE;  
CHANNEL2 DONTCARE;CHANNEL3 DONTCARE;  
CHANNEL4 DONTCARE;CLOCK:SOURCE NONE;;  
TRIGGER:SIMPLE:PATTERN:CONDITION ENTER;;  
TRIGGER:SIMPLE:WIDTH:CONDITION IN;  
TIME 5.0E-09;SOURCE 1;SLOPE HIGH;;  
TRIGGER:SIMPLE:DELAY 0.0E+00;HOLDOFF:  
MODE 0;TIME 0.0E+00

**TRIGger[:SIMple]:DELay**

**Function** Sets the delay time (Normal delay) for trigger, or queries the current setting.

**Syntax** TRIGger[:SIMple]:DELay <Time>  
TRIGger[:SIMple]:DELay?  
<Time>= For a description of the setting procedure, refer to User's Manual IM701530-01E.

**Example** TRIGGER:SIMPLE:DELAY 0.0  
TRIGGER:SIMPLE:DELAY?→:TRIGGER:SIMPLE:  
DELAY 0.0E+00

**TRIGger[:SIMple]:EDGE?**

**Function** Queries all the settings when an edge trigger is selected.

**Syntax** TRIGger[:SIMple]:EDGE?

**Example** TRIGGER:SIMPLE:EDGE?→:TRIGGER:SIMPLE:  
EDGE:SOURCE 1;SLOPE RISE

**TRIGger[:SIMple]:EDGE:SLOPe**

**Function** Sets the trigger slope when an edge trigger is selected, or queries the current setting.

**Syntax** TRIGger[:SIMple]:EDGE:SLOPe  
{BOTH|FALL|RISE}  
TRIGger[:SIMple]:EDGE:SLOPe?

**Example** TRIGGER:SIMPLE:EDGE:SLOPE RISE  
TRIGGER:SIMPLE:EDGE:SLOPE?→:TRIGGER:  
SIMPLE:EDGE:SLOPE RISE

**TRIGger[:SIMple]:EDGE:SOURce**

**Function** Sets the trigger source when an edge trigger is selected, or queries the current setting.

**Syntax** TRIGger[:SIMple]:EDGE:SOURce  
{<NRf>}|EXTERNAL|LINE  
TRIGger[:SIMple]:EDGE:SOURce?  
{<NRf>}=1 to 4

**Example** TRIGGER:SIMPLE:EDGE:SOURCE 1  
TRIGGER:SIMPLE:EDGE:SOURCE?→:TRIGGER:  
SIMPLE:EDGE:SOURCE 1

**TRIGger[:SIMple]:FUNction**

**Function** Sets the type of trigger, or queries the current setting.

**Syntax** TRIGger[:SIMple]:FUNction  
{EDGE|OR|PATTERN|TV|WIDTH|WINDOW}  
TRIGger[:SIMple]:FUNction?

**Example** TRIGGER:SIMPLE:FUNCTION EDGE  
TRIGGER:SIMPLE:FUNCTION?→:TRIGGER:  
SIMPLE:FUNCTION EDGE

**TRIGger[:SIMple]:HOLDoff?**

**Function** Queries all the hold-off settings for a trigger.

**Syntax** TRIGger[:SIMple]:HOLDoff?

**Example** TRIGGER:SIMPLE:HOLDOFF?→:TRIGGER:SIMPLE:  
HOLDOFF:MODE 0;TIME 200.0E-09

**TRIGger[:SIMPlE:]HOLDoff:MODE**

**Function** Turns hold-off ON/OFF for a trigger, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:HOLDoff:MODE {<Boolean>}  
TRIGger[:SIMPlE]:HOLDoff:MODE?

**Example** TRIGGER:SIMPLE:HOLDOFF:MODE OFF  
TRIGGER:SIMPLE:HOLDOFF:MODE?→:TRIGGER:  
SIMPLE:HOLDOFF:MODE 0

**TRIGger[:SIMPlE:]HOLDoff:TIME**

**Function** Sets the hold-off time for a trigger, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:HOLDoff:TIME <Time>  
TRIGger[:SIMPlE]:HOLDoff:TIME?  
<Time>=For a description of the setting procedure, refer to User's Manual IM701530-01E.

**Example** TRIGGER:SIMPLE:HOLDOFF:TIME 200NS  
TRIGGER:SIMPLE:HOLDOFF:TIME?→:TRIGGER:  
SIMPLE:HOLDOFF:TIME 200.0E-09

**TRIGger[:SIMPlE:]OR?**

**Function** Queries all the settings when an OR trigger is selected.

**Syntax** TRIGger[:SIMPlE]:OR?

**Example** TRIGGER:SIMPLE:OR?→:TRIGGER:SIMPLE:OR:  
CHANNEL1 DONTCARE;CHANNEL2 DONTCARE;  
CHANNEL3 DONTCARE;CHANNEL4 DONTCARE

**TRIGger[:SIMPlE:]OR:CHANnel<x>**

**Function** Sets the slope and state when an OR trigger is selected, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:OR:CHANnel<x>  
{DONTcare|FALL|RISE}  
TRIGger[:SIMPlE]:OR:CHANnel<x>?  
<x>=1 to 4

**Example** TRIGGER:SIMPLE:OR:CHANNEL1 DONTCARE  
TRIGGER:SIMPLE:OR:CHANNEL1?→:TRIGGER:  
SIMPLE:OR:CHANNEL1 DONTCARE

**TRIGger[:SIMPlE:]PATTern?**

**Function** Queries all the settings when a pattern trigger is selected as the trigger.

**Syntax** TRIGger[:SIMPlE]:PATTern?

**Example** TRIGGER:SIMPLE:PATTERN?→:TRIGGER:  
SIMPLE:PATTERN:CHANNEL1 DONTCARE;  
CHANNEL2 DONTCARE;CHANNEL3 DONTCARE;  
CHANNEL4 DONTCARE;CLOCK:SOURCE NONE;;  
TRIGGER:SIMPLE:PATTERN:CONDITION ENTER

**TRIGger[:SIMPlE:]PATTern:CHANnel<x>**

**Function** Sets the state for the specified channel when a pattern trigger is selected, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:PATTern:  
CHANnel<x> {DONTcare|HIGH|LOW}  
TRIGger[:SIMPlE]:PATTern:CHANnel<x>?  
<x>=1 to 4

**Example** TRIGGER:SIMPLE:PATTERN:CHANNEL1 DONTCARE  
TRIGGER:SIMPLE:PATTERN:CHANNEL1?→:  
TRIGGER:SIMPLE:PATTERN:CHANNEL1 DONTCARE

**TRIGger[:SIMPlE:]PATTern:CLOCK?**

**Function** Queries all the clock channel settings when a pattern trigger is selected.

**Syntax** TRIGger[:SIMPlE]:PATTern:CLOCK?

**Example** TRIGGER:SIMPLE:PATTERN:CLOCK?→:TRIGGER:  
SIMPLE:PATTERN:CLOCK:SOURCE 1:SLOPE RISE

**TRIGger[:SIMPlE:]PATTern:CLOCK:SLOPe**

**Function** Sets the slope for the clock channel when a pattern trigger is selected, or queries the current setting. A command other than "TRIGger[:SIMPlE]:PATTern:CLOCK:SOURCE NONE" must be used, otherwise the setting/query is meaningless.

**Syntax** TRIGger[:SIMPlE]:PATTern:CLOCK:  
SLOPe{FALL|RISE}  
TRIGger[:SIMPlE]:PATTern:CLOCK:SLOPe?

**Example** TRIGGER:SIMPLE:PATTERN:CLOCK:SLOPE RISE  
TRIGGER:SIMPLE:PATTERN:CLOCK:SLOPE?→:  
TRIGGER:SIMPLE:PATTERN:CLOCK:SLOPE RISE

**TRIGger[:SIMPlE:]PATTern:CLOCK:SOURCE**

**Function** Sets the clock channel when a pattern trigger is selected, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:PATTern:CLOCK:  
SOURCE {<NRF>}|NONE}  
TRIGger[:SIMPlE]:PATTern:CLOCK:SOURCE?  
{<NRF>}=1 to 4

**Example** TRIGGER:SIMPLE:PATTERN:CLOCK:SOURCE 1  
TRIGGER:SIMPLE:PATTERN:CLOCK:SOURCE?→:  
TRIGGER:SIMPLE:PATTERN:CLOCK:SOURCE 1

**TRIGger[:SIMPlE:]PATTern:CONDition**

**Function** Sets the trigger condition when a pattern trigger is selected, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:PATTern:  
CONDition {ENTER|EXIT|FALSE|TRUE}  
TRIGger[:SIMPlE]:PATTern:CONDition?

**Example** TRIGGER:SIMPLE:PATTERN:CONDITION ENTER  
TRIGGER:SIMPLE:PATTERN:CONDITION?→:  
TRIGGER:SIMPLE:PATTERN:CONDITION ENTER

**Description** Select "FALSE" or "TRUE" if "TRIGger[:SIMPlE]:PATTern:CLOCK:SOURCE NONE" has been selected, or select "ENTER" or "EXIT" if another command has been selected.

**TRIGger[:SIMPlE:]TV?**

**Function** Queries all the settings when a TV trigger has been selected.

**Syntax** TRIGger[:SIMPlE]:TV?

**Example** TRIGGER:SIMPLE:TV?→:TRIGGER:SIMPLE:TV:  
TYPE NTSC;FIELD 1;LINE 5;  
POLARITY NEGATIVE;LEVEL 0.50;FRAME 1

**TRIGger[:SIMPlE]:TV:FIELD**

**Function** Sets the field No. when a TV trigger has been selected, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:TV:FIELD  
{<NRf>}|DONTcare}  
TRIGger[:SIMPlE]:TV:FIELD?  
{<NRf>}=1, 2

**Example** TRIGGER:SIMPLE:TV:FIELD 1  
TRIGGER:SIMPLE:TV:FIELD?→:TRIGGER:  
SIMPLE:TV:FIELD 1

**TRIGger[:SIMPlE]:TV:FRAME**

**Function** Sets the frame skip function when the trigger is set to TV trigger, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:TV:FRAME {<NRf>}  
TRIGger[:SIMPlE]:TV:FRAME?  
{<NRf>}=1, 2, 4, 8

**Example** TRIGGER:SIMPLE:TV:FRAME 2  
TRIGGER:SIMPLE:TV:FRAME?→:  
TRIGGER[:SIMPlE]:TV:FRAME 2

**TRIGger[:SIMPlE]:TV:LEVEl**

**Function** Sets the trigger level when a TV trigger is selected, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:TV:LEVEl {<NRf>}  
TRIGger[:SIMPlE]:TV:LEVEl?  
{<NRf>}=0.10 to 2.00(in 0.05 div steps)

**Example** TRIGGER:SIMPLE:TV:LEVEL 0.5  
TRIGGER:SIMPLE:TV:LEVEL?→:TRIGGER:  
SIMPLE:TV:LEVEL 0.50

**TRIGger[:SIMPlE]:TV:LINE**

**Function** Sets the line No. when a TV trigger is selected, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:TV:LINE {<NRf>}  
TRIGger[:SIMPlE]:TV:LINE?  
{<NRf>}=5 to 1054(NTSC)  
2 to 1251(PAL)  
2 to 2251(HDTV)

**Example** TRIGGER:SIMPLE:TV:LINE 5  
TRIGGER:SIMPLE:TV:LINE?→:TRIGGER:SIMPLE:  
TV:LINE 5

**TRIGger[:SIMPlE]:TV:POLarity**

**Function** Sets the polarity when a TV trigger is selected, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:TV:POLarity{POSitive|  
NEGative}  
TRIGger[:SIMPlE]:TV:POLarity?

**Example** TRIGGER:SIMPLE:TV:POLARITY NEGATIVE  
TRIGGER:SIMPLE:TV:POLARITY?→:TRIGGER:  
SIMPLE:TV:POLARITY NEGATIVE

**TRIGger[:SIMPlE]:TV:TYPE**

**Function** Sets the broadcasting system when a TV trigger is selected, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:TV:TYPE {HDTV|NTSC|PAL}  
TRIGger[:SIMPlE]:TV:TYPE?

**Example** TRIGGER:SIMPLE:TV:TYPE NTSC  
TRIGGER:SIMPLE:TV:TYPE?→:TRIGGER:SIMPLE:  
TV:TYPE NTSC

**TRIGger[:SIMPlE]:WIDTH?**

**Function** Queries all the settings when a pulse trigger is selected.

**Syntax** TRIGger[:SIMPlE]:WIDTH?

**Example** TRIGGER:SIMPLE:WIDTH?→:TRIGGER:SIMPLE:  
WIDTH:CONDITION IN;TIME 5.0E-09;  
SOURCE 1;SLOPE HIGH

**TRIGger[:SIMPlE]:WIDTH:CONDition**

**Function** Sets the trigger condition (IN/OUT) when a pulse trigger has been selected, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:WIDTH:CONDition{IN|OUT|  
TIMEout}  
TRIGger[:SIMPlE]:WIDTH:CONDition?

**Example** TRIGGER:SIMPLE:WIDTH:CONDITION IN  
TRIGGER:SIMPLE:WIDTH:CONDITION?→:  
TRIGGER:SIMPLE:WIDTH:CONDITION IN

**TRIGger[:SIMPlE]:WIDTH:SLOPe**

**Function** Sets the trigger state when a pulse trigger is selected, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:WIDTH:SLOPe {HIGH|LOW}  
TRIGger[:SIMPlE]:WIDTH:SLOPe?

**Example** TRIGGER:SIMPLE:WIDTH:SLOPE HIGH  
TRIGGER:SIMPLE:WIDTH:SLOPE?→:TRIGGER:  
SIMPLE:WIDTH:SLOPE HIGH

**TRIGger[:SIMPlE]:WIDTH:SOURce**

**Function** Sets the trigger source when a pulse trigger is selected, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:WIDTH:SOURce {<NRf>}  
TRIGger[:SIMPlE]:WIDTH:SOURce?  
{<NRf>}=1 to 4

**Example** TRIGGER:SIMPLE:WIDTH:SOURCE 1  
TRIGGER:SIMPLE:WIDTH:SOURCE?→:TRIGGER:  
SIMPLE:WIDTH:SOURCE 1

**TRIGger[:SIMPlE]:WIDTH:TIME**

**Function** Sets the pulse width when a pulse trigger is selected, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:WIDTH:TIME <Time>  
TRIGger[:SIMPlE]:WIDTH:TIME?  
<Time>=the smallest setting value in case  
of trigger condition IN is 5ns,  
whereas this value is 30ns in case  
of trigger condition OUT and  
TIMEOUT.

For a description of the setting  
procedure, refer to User's Manual  
IM701530-01E.

**Example** TRIGGER:SIMPLE:WIDTH:TIME 10NS  
TRIGGER:SIMPLE:WIDTH:TIME?→:TRIGGER:  
SIMPLE:WIDTH:TIME 10.0E-09

**TRIGger[:SIMPlE]:WINDow?**

**Function** Queries all settings relating to the window trigger.

**Syntax** TRIGger[:SIMPlE]:WINDow?

**Example** TRIGGER:SIMPLE:WINDOW?→:TRIGGER:SIMPLE:  
WINDOW:CONDITION IN;LEVEL 0.0E+00;  
WIDTH 50.0E+00

**TRIGger[:SIMPlE]:WINDow:CONDition**

**Function** Sets the trigger condition when a window trigger has been selected, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:WINDow:CONDition{IN|OUT}  
TRIGger[:SIMPlE]:WINDow:CONDition?

**Example** TRIGGER:SIMPLE:WINDOW:CONDITION IN  
TRIGGER:SIMPLE:WINDOW:CONDITION?→:  
TRIGGER:SIMPLE:WINDOW:CONDITION IN

**TRIGger[:SIMPlE]:WINDow:LEVel**

**Function** Sets the center level for a window trigger, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:WINDow:LEVel {<NRf>}  
TRIGger[:SIMPlE]:WINDow:LEVel?  
{<NRf>}=±10div (in steps of 1/50 of the  
V/div value)

**Example** TRIGGER:SIMPLE:WINDOW:LEVEL 50V  
TRIGGER:SIMPLE:WINDOW:LEVEL?→:TRIGGER:  
SIMPLE:WINDOW:LEVEL 50.0E+00

**TRIGger[:SIMPlE]:WINDow:WIDTh**

**Function** Sets the width of the window for a window trigger, or queries the current setting.

**Syntax** TRIGger[:SIMPlE]:WINDow:WIDTh {<NRf>}  
TRIGger[:SIMPlE]:WINDow:WIDTh?  
{<NRf>}=±10div (in steps of 1/50 of the  
V/div value)

**Example** TRIGGER:SIMPLE:WINDOW:WIDTH 50V  
TRIGGER:SIMPLE:WINDOW:WIDTH?→:TRIGGER:  
SIMPLE:WINDOW:WIDTH 50.0E+00

**TRIGger:SOURce?**

**Function** Queries all the settings relating to the trigger source.

**Syntax** TRIGger:SOURce?

**Example** TRIGGER:SOURCE?→:TRIGGER:SOURCE:  
COUPLING AC;HFREJECTION 0;CHANNEL1:  
LEVEL 0.0E+00;:TRIGGER:SOURCE:CHANNEL2:  
LEVEL 0.0E+00;:TRIGGER:SOURCE:CHANNEL3:  
LEVEL 0.0E+00;:TRIGGER:SOURCE:CHANNEL4:  
LEVEL 0.0E+00;:TRIGGER:SOURCE:EXTERNAL:  
LEVEL 150.0E-03

**TRIGger:SOURce:CHANnel<x>?**

**Function** Queries all the trigger source settings for the specified channel.

**Syntax** TRIGger:SOURce:CHANnel<x>?  
<x>=1 to 4

**Example** TRIGGER:SOURCE:CHANNEL1?→:TRIGGER:  
SOURCE:CHANNEL1:LEVEL 0.0E+00

**TRIGger:SOURce:CHANnel<x>:LEVel**

**Function** Sets the trigger level (or the center level in case of the OR or pulse trigger), or queries the current setting.

**Syntax** TRIGger:SOURce:CHANnel<x>:  
LEVel {<Voltage>}  
TRIGger:SOURce:CHANnel<x>:LEVel?  
<x>=1 to 4  
<Voltage>=±10div (in steps of 1/50 of the  
V/div value)

**Example** TRIGGER:SOURCE:CHANNEL1:LEVEL 0V  
TRIGGER:SOURCE:CHANNEL1:LEVEL?→:TRIGGER:  
SOURCE:CHANNEL1:LEVEL 0.0E+00

**Description** For a TV trigger, the setting must be made using  
“TRIGger[:SIMPlE]:TV:LEVel”.

**TRIGger:SOURce:COUPling**

**Function** Sets the trigger coupling, or queries the current setting.

**Syntax** TRIGger:SOURce:COUPling {AC|DC}  
TRIGger:SOURce:COUPling?

**Example** TRIGGER:SOURCE:COUPLING AC  
TRIGGER:SOURCE:COUPLING?→:TRIGGER:  
SOURCE:COUPLING AC

**TRIGger:SOURce:EXTernal?**

**Function** Queries the external trigger settings.

**Syntax** TRIGger:SOURce:EXTernal?

**Example** TRIGGER:SOURCE:EXTERNAL?→:TRIGGER:  
SOURCE:EXTERNAL LEVEL 1.5E+00

**TRIGger:SOURce:EXTernal:LEVel**

**Function** Sets the external trigger level, or queries the current setting.

**Syntax** TRIGger:SOURce:EXTernal:LEVel <Voltage>  
TRIGger:SOURce:EXTernal:LEVel?  
<Voltage>=0.15V or 1.5V

**Example** TRIGGER:SOURCE:EXTERNAL:LEVEL 1.5V  
TRIGGER:SOURCE:EXTERNAL:LEVEL→:TRIGGER:  
SOURCE:EXTERNAL:LEVEL 1.5E+00

**TRIGger:SOURce:HFRejection**

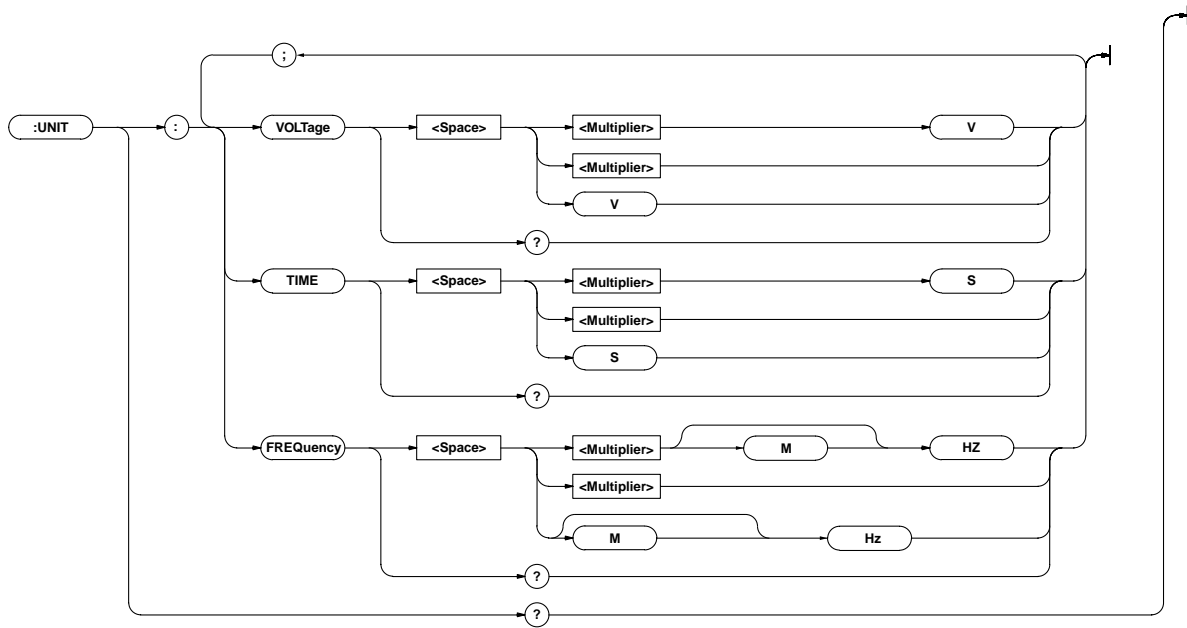
**Function** Turns the HF rejection ON/OFF, or queries the current setting.

**Syntax** TRIGger:SOURce:HFRejection {<Boolean>}  
TRIGger:SOURce:HFRejection?

**Example** TRIGGER:SOURCE:HFREJECTION OFF  
TRIGGER:SOURCE:HFREJECTION?→:TRIGGER:  
SOURCE:HFREJECTION 0

## 4.28 UNIT Group

The commands in the UNIT group are used to make settings and inquiries about the default values for <Voltage>, <Time> and <Frequency>. There is no front panel key for this operation.



### UNIT?

**Function** Queries the default values of <Voltage>, <Time> and <Frequency>.

**Syntax** UNIT?

**Example** UNIT?→:UNIT:FREQUENCY HZ;TIME S;  
VOLTAGE V

### UNIT:FREQuency

**Function** Sets the default value for <Frequency>, or queries the current setting.

**Syntax** UNIT:FREQuency {<Multiplier>[M]HZ|  
<Multiplier>|[M]HZ}  
UNIT:FREQuency?  
<Multiplier>= Refer to page 3-5.

**Example** UNIT:FREQUENCY HZ  
UNIT:FREQUENCY?→:UNIT:FREQUENCY HZ

### UNIT:TIME

**Function** Sets the default value for <Time>, or queries the current setting.

**Syntax** UNIT:TIME {<Multiplier>S|<Multiplier>|S}  
UNIT:TIME?  
<Multiplier>= Refer to page 3-5.

**Example** UNIT:TIME S  
UNIT:TIME?→:UNIT:TIME S

### UNIT:VOLTage

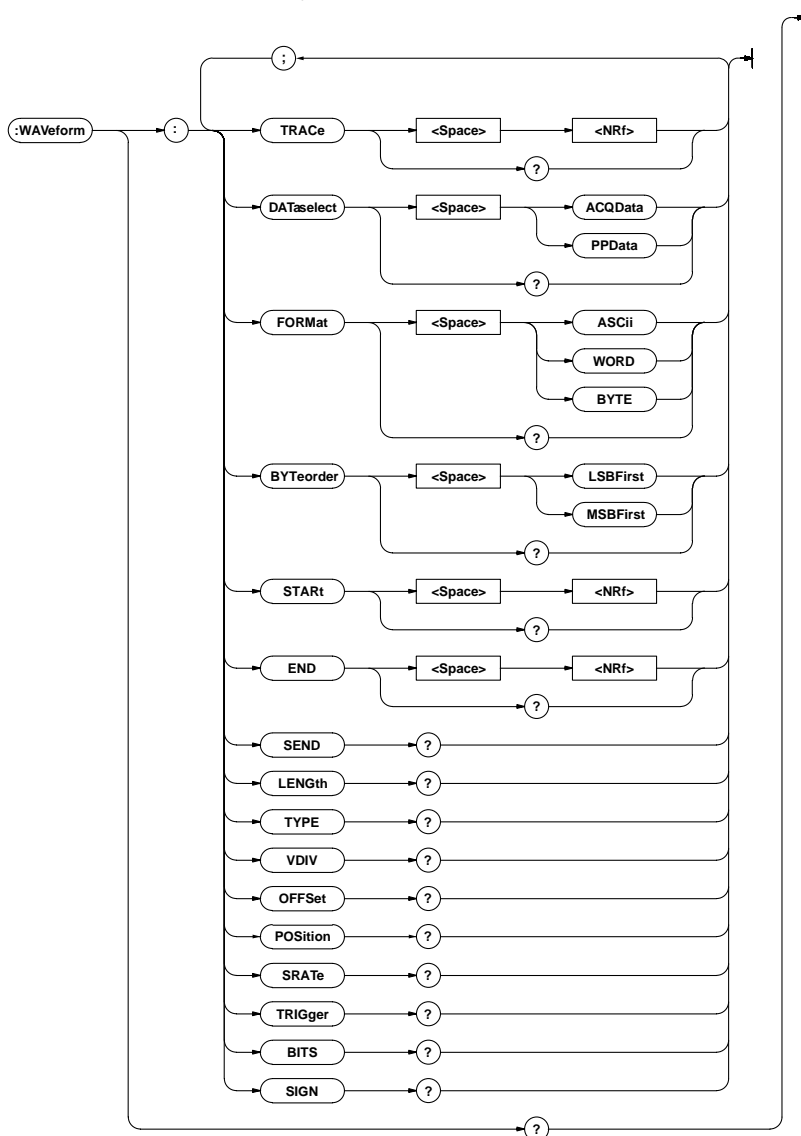
**Function** Sets the default value for <Voltage>, or queries the current setting.

**Syntax** UNIT:VOLTage {<Multiplier>V|<Multiplier>|  
V}  
UNIT:VOLTage?  
<Multiplier>= Refer to page 3-5.

**Example** UNIT:VOLTAGE V  
UNIT:VOLTAGE?→:UNIT:VOLTAGE V

## 4.29 WAVEform Group

The commands in the WAVEform group are used to make settings relating to, and inquiries about acquired waveform data. There is no front panel key for this operation.



### WAVEform?

**Function** Queries all information relating to the waveform data.

**Syntax** WAVEform?

**Example** WAVEFORM?→:WAVEFORM:TRACE 1;FORMAT WORD; BYTEORDER MSBFIRST;DATASELECT ACQDATA; START 0;END 10031

### WAVEform:BITS?

**Function** Queries the bit length of the waveform data as designated by "WAVEform:TRACE" including the bit corresponding to the sign of the waveform data.

**Syntax** WAVEform:BITS?

**Example** WAVEFORM:BITS?→:WAVEFORM:BITS 9

**Description** The oscilloscope will always return either "8" or "9".

### WAVEform:BYTeorder

**Function** Sets the transmission order when words consist of more than 2 bytes, or queries the current setting.

**Syntax** WAVEform:BYTeorder {LSBFirst|MSBFirst}  
WAVEform:BYTeorder?

**Example** WAVEFORM:BYTEORDER MSBFIRST  
WAVEFORM:BYTEORDER?→:WAVEFORM:  
BYTEORDER MSBFIRST

### WAVEform:DATaselect

**Function** Selects whether acquisition data designated by "WAVEform:TRACE" is ACQ waveform data or P-P waveform data, or queries the current setting.

**Syntax** WAVEform:DATaselect {ACQData|PPData}  
WAVEform:DATaselect?

**Example** WAVEFORM:DATASELECT ACQDATA  
WAVEFORM:DATASELECT?→:WAVEFORM:  
DATASELECT ACQDATA



**WAVEform:END**

**Function** Sets the point at which the last item of data of the waveform designated by "WAVEform:TRACe" is to be located, or queries the current setting.

**Syntax** WAVEform:END {<NRf>}  
WAVEform:END?  
{<NRf>}=0 to 120255 (for DL1540C)  
0 to 2004015 (for DL1540CL)

**Example** WAVEFORM:END 120255  
WAVEFORM:END?→:WAVEFORM:END 120255

**WAVEform:FORMat**

**Function** Selects the format in which the data is to be transmitted, or queries the current setting.

**Syntax** WAVEform:FORMat {ASCIi|BYTE|WORD}  
WAVEform:FORMat?

**Example** WAVEFORM:FORMAT ASCII  
WAVEFORM:FORMAT?→:WAVEFORM:FORMAT ASCII

**Description** • If the data format is "ASCIi", a response to "WAVEform:SEND?" will be returned as a sequence <Voltage>, <Voltage>, etc.  
• If the data format is "BYTE", a response to "WAVEform:SEND?" will be returned as <Block data>. The voltage conversion expression varies according to whether data is signed or unsigned. You can use the WAVEform:SIGN? command to determine whether the data is signed. Conversion to a voltage value is possible using the following equation.

(1) If data is signed:

$$\text{Voltage} = \frac{V/\text{div} \times \text{Byte data}}{12.5} + \text{Offset}$$

"WAVEform:BITS?" will return "9".

(2) If data is unsigned:

$$\text{Voltage} = \frac{V/\text{div} \times \{\text{Byte data-Position}\}}{25} + \text{Offset}$$

"WAVEform:BITS?" will return "8".

• If the data format is "WORD", a response to "WAVEform:SEND?" will be returned as <block data>. Conversion to a voltage value is possible using the following equation.

$$\text{Voltage} = \frac{V/\text{div} \times \text{Word data}}{3200} + \text{Offset}$$

Word data is always signed.

• Designation as byte or word can be made irrespective of the original length in bits of the waveform data. If the designated data length differs from the original data length (which can be verified using the "WAVEform:BITS?" command), the data type will be extended using extra bits or reduced by discarding the least significant bits.

**WAVEform:LENGth?**

**Function** Returns the number of records in the waveform designated by "WAVEform:TRACe".

**Syntax** WAVEform:LENGth?

**Example** WAVEFORM:LENGTH?→:WAVEFORM:LENGTH 120256

**Description** For DL1540C, the returned data length will lie within 1 to 120256, for DL1540CL, the returned data length will lie within 1 to 2004016 (1002 in case of displayed waveform data).

**WAVEform:OFFSet?**

**Function** Queries the offset voltage of the waveform designated by "WAVEform:TRACe".

**Syntax** WAVEform:OFFSet?

**Example** WAVEFORM:OFFSET?→:WAVEFORM:  
OFFSET 0.0000E+00

**WAVEform:POSition?**

**Function** Where waveform data specified by the WAVEform:TRACe command in unsigned, this command sets the vertical-axis position used for the voltage-conversion calculation.

**Syntax** WAVEform:POSition?

**Example** WAVEFORM:POSITION?→:WAVEFORM:  
POSITION 1.00

**WAVEform:SEND?**

**Function** Queries the waveform data designated by "WAVEform:TRACe".

**Syntax** WAVEform:SEND?

**Example** WAVEFORM:SEND?→#6(bytes of 6digits)(Data  
byte string)(Block data) (for DL1540C)  
WAVEFORM:SEND?→#8(bytes of 8digits)(Data  
byte string)(Block data) (for DL1540CL)  
or, <Voltage>, <Voltage>...

**Description** A response to this command is returned as <Block data> if "WORD" or "BYTE" is selected for "WAVEform:FORMat"; or as a sequence <voltage>, etc. if "ASCIi" is selected. On the DL1540C, number of bytes in <block data> is (2 + 6 + number of data + 1 (delimiter)). On the DL1540CL, number of bytes in <block data> is (2 + 8 + number of data + 1 (delimiter)). For information about block data, refer to page 3-6.

**WAVEform:SIGN?**

**Function** Queries whether the data is signed or unsigned when binary data is used to make a query about the target waveform designated by "WAVEform:TRACe".

**Syntax** WAVEform:SIGN?

**Example** WAVEFORM:SIGN?→:WAVEFORM:SIGN 1

**Description** "1" is returned if the data is signed. "0" is returned if the data is unsigned.

**WAVEform:SRATe?**

**Function** Queries the sample rate of the waveform.

**Syntax** WAVEform:SRATe?

**Example** WAVEFORM:SRATE?→:WAVEFORM:  
SRATE 1.0000E+06

**Description** The same query can be made using "TIMebase:SRATe?".

**WAVEform:START**

**Function** Sets the point at which the first item of data of the waveform designated by "WAVEform:TRACe" is to be located, or queries the current setting.

**Syntax** WAVEform:START {<NRf>}  
WAVEform:START?  
{<NRf>}=0 to 120255 (for DL1540C)  
0 to 2004015 (for DL1540CL)

**Example** WAVEFORM:START 0  
WAVEFORM:START?→:WAVEFORM:START 0

**WAVEform:TRACe**

**Function** Set the target waveform for the WAVEform group, or queries the current setting.

**Syntax** WAVEform:TRACe {<NRf>}  
WAVEform:TRACe?  
{<NRf>}=1 to 4

**Example** WAVEFORM:TRACE 1  
WAVEFORM:TRACE?→:WAVEFORM:TRACE 1

**WAVEform:TRIGger?**

**Function** Queries the points elapsed from the start point designated by "WAVEform:STARt" until the trigger point.

**Syntax** WAVEform:TRIGger?

**Example** WAVEFORM:TRIGGER?→:WAVEFORM:TRIGGER 2000

**Description** The data number of the trigger point will be returned. In case the trigger point is located before the starting point, a minus value will be returned.

**WAVEform:TYPE?**

**Function** Queries the type of ACQ waveform data.

**Syntax** WAVEform:TYPE?

**Example** WAVEFORM:TYPE?→:WAVEFORM:TYPE NORMAL

**Description** Any one from {AVERAge|ENVELOpe|NORMAl} will be returned.

**WAVEform:VDIV?**

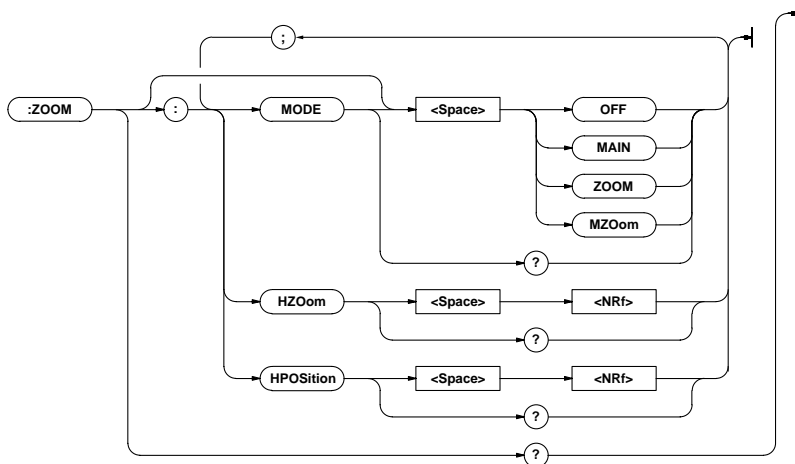
**Function** Queries the V/div value for the waveform designated by "WAVEform:TRACe".

**Syntax** WAVEform:VDIV?

**Example** WAVEFORM:VDIV?→:WAVEFORM:VDIV 50.0E+00

**4.30 ZOOM Group**

The commands in the ZOOM group are used for making settings relating to, and inquiries about waveform expansion. This allows you to make the same settings as when using the ZOOM key on the front panel.



**ZOOM?**

**Function** Queries all the settings relating to waveform expansion.

**Syntax** ZOOM?

**Example** ZOOM?→:ZOOM:MODE MAIN;HZOOM 1.0E+00;  
HPOSITION 0.0000

**ZOOM:HPOSITION**

**Function** Sets the center position for time axis expansion, or queries the current setting.

**Syntax** ZOOM:HPOSITION {<NRf>}  
ZOOM:HPOSITION?  
{<NRf>}=-5.0000 to 5.0000(in steps of  
0.0001div)

**Example** ZOOM:HPOSITION 0  
ZOOM:HPOSITION?→:ZOOM:HPOSITION 0.0000

**ZOOM:HZOOM**

**Function** Sets the expansion rate in the horizontal direction (time axis), or queries the current setting.

**Syntax** ZOOM:HZOOM {<NRf>}  
{<NRf>}=1 to 10000

**Example** ZOOM:HZOOM 1  
ZOOM:HZOOM?→:ZOOM:HZOOM 1.0E+00

**ZOOM[:MODE]**

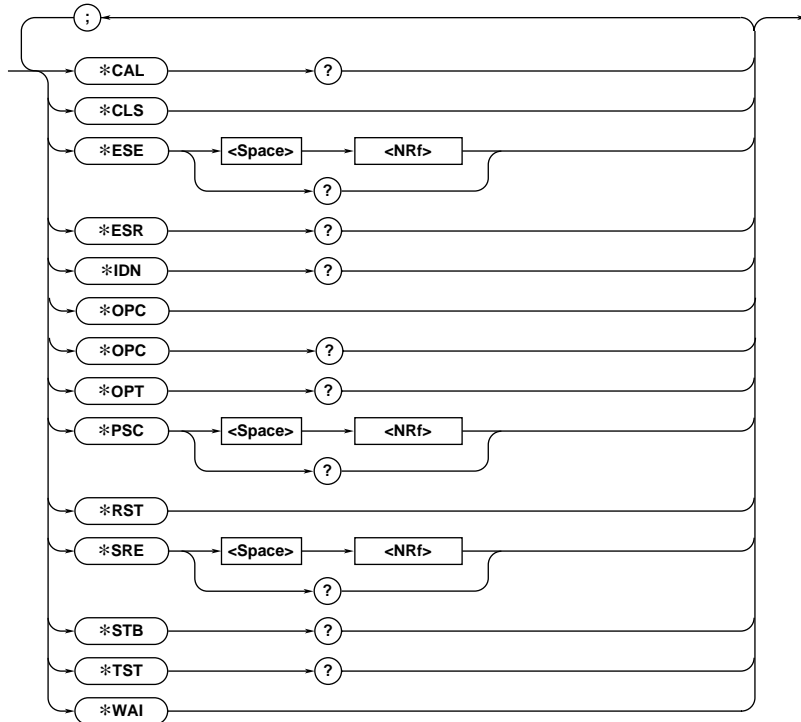
**Function** Sets the display type for the expanded waveform, or queries the current setting.

**Syntax** ZOOM[:MODE] {OFF|MAIN|MZOOM|ZOOM}  
ZOOM:MODE?

**Example** ZOOM:MODE MAIN  
ZOOM:MODE?→:ZOOM:MODE MAIN

### 4.31 Common Command Group

The commands in the common command group are independent of the instrument's functions, and are specified in IEEE 488.2-1987. There is no front panel key that corresponds to this group.



#### \*CAL?

**Function** Performs calibration and queries the result.

**Syntax** \*CAL?

**Example** \*CAL?

**Description** "0" will be returned if calibration has been performed properly. "1" will be returned if an abnormality has been detected during calibration.

#### \*CLS

**Function** Clears the standard event register, extended event register and error queue.

**Syntax** \*CLS

**Example** \*CLS

**Description**

- The output will also be cleared if a \*CLS command is appended after the program message terminator.
- For details of the registers and queues, refer to chapter 5.

#### \*ESE

**Function** Sets the value for the standard event enable register, or queries the current setting.

**Syntax** \*ESE {<NRf>}

\*ESE?

{<NRf>}=0 to 255

**Example** \*ESE 251

\*ESE?→251

**Description** • <NRf> is the sum of the bits expressed as a decimal number.

- For example, if "\*ESE 251" is set, the standard event enable register will be set to "11111011". This means that bit 2 of the standard event register is disabled so that bit 5 (ESB) of the status byte register will not be set to "1", even if a query error occurs.
- Default is "\*ESE 0", i.e. all bits are disabled.
- The standard event enable register will not be cleared, even if an inquiry is made using "\*ESE?".
- For details of the standard event enable register, refer to page 5-2.

**\*ESR?**

**Function** Queries the value of the standard event register and clears it at the same time.

**Syntax** \*ESR?

**Example** \*ESR?→32

**Description**

- <NRf> is the sum of the bits expressed as a decimal number.
- It is possible to ascertain the type of event which has occurred, while SRQ is occurring.
- For example, if “\*ESR 32” is returned, this means that the standard event register is “00100000”, i.e. the SRQ has occurred due to a command syntax error.
- If a query is made using “\*ESR?”, the standard event register will be cleared.
- For details of the standard event register, refer to page 5-3.

**\*IDN?**

**Function** Queries the instrument model.

**Syntax** \*IDN?

**Example** \*IDN?→YOKOGAWA,701530,0,F1.01

**Description** A reply consists of the following sequence: <Model>,<Serial No.> and <Firmware version>.

**\*OPC**

**Function** After “\*OPC” is sent, this command sets bit 0 (the OPC bit) of the standard event register to “1” when execution of the specified overlap command has been completed.

**Syntax** \*OPC

**Example** \*OPC

**Description**

- For a Description of the synchronization method using “\*OPC”, refer to page 3-7.
- Designation of an overlap command is performed using “COMMunicate:OPSE”.

**\*OPC?**

**Function** After “\*OPC?” is sent, “1” (ASCII) will be returned if execution of the designated overlap command has been completed.

**Syntax** \*OPC?

**Example** \*OPC?→1

**Description**

- For a Description of the synchronization method using “\*OPC”, refer to page 3-7.
- Designation of an overlap command is performed using “COMMunicate:OPSE”.

**\*OPT?**

**Function** Queries installed options.

**Syntax** \*OPT?

**Example** \*OPT?→PRINTER

**Description**

- “NONE” will be attached to the reply if no options are installed.
- “OPT?” must always be the last query in a program message. If there is another query after this, an error will occur.

**\*PSC**

**Function** Selects whether or not to clear the following registers when power is turned ON, or queries the current setting. The registers are the standard event enable register, the extended event enable register and the transition filter. However, they cannot be cleared if the parameter is “0”.

**Syntax** \*PSC {<NRf>}

\*PSC?

{<NRf>}=0 (does not clear the registers, a value other than 0 clears the registers)

**Example** \*PSC 1  
\*PSC?→1

**Description** For details of each register, refer to chapter 5.

**\*RST**

**Function** Resets the current settings.

**Syntax** \*RST

**Example** \*RST

**Description** “\*OC” and “\*OPC?” will also be reset.

**\*SRE**

**Function** Sets the value of the service request enable register, or queries the current setting.

**Syntax** \*SRE {<NRf>}

\*SRE?

{<NRf>}=0 to 255

**Example** \*SRE 239  
\*SRE?→191

**Description**

- <NRf> is the sum of the bits expressed as a decimal number.
- For example, if “\*SRE 239” is set, the service request enable register will be set to “11101111”. This means that bit 4 of the service request enable register is disabled, so that bit 5 (ESB) of the status byte register will not be set to “1”, even if the output queue is not empty.
- However, bit 6 (MSS) of the status byte register is the MSS bit, so it will be ignored.
- Default is “\*SRE 0”, i.e. all bits are disabled.
- The service request enable register will not be cleared, even if a query is made using “\*SRE?”.
- For details of the service request enable register, refer to page 5-2.

## 4.31 Common Command Group

---

### \*STB?

**Function** Queries the value of the status byte register.

**Syntax** \*STB?

**Example** \*STB?→4

**Description**

- The sum of the bits expressed as a decimal number is returned.
- Bit 6 is MSS not RQS, since the register is read without serial polling.
- For example, if “\*STB 4” is returned, the status byte register is set to “00000100”, i.e. the error queue is not empty (an error has occurred).
- The status byte register will be cleared, even if a query is made using “\*STB?”.
- For details on the status byte register, refer to page 5-2.

### \*TST?

**Function** Executes a self-test and queries the result. All internal memory boards are tested.

**Syntax** \*TST?

**Example** \*TST?→0

**Description**

- “0” will be returned if the self-test result is satisfactory. Value other than “0” will be returned if an abnormality is detected during the test.
- The same function can be performed using “SELFtest:BOARD:EXECute?”.

### \*WAI

**Function** Waits for the command following “\*WAI” until execution of the designated overlap command has been completed.

**Syntax** \*WAI

**Example** \*WAI

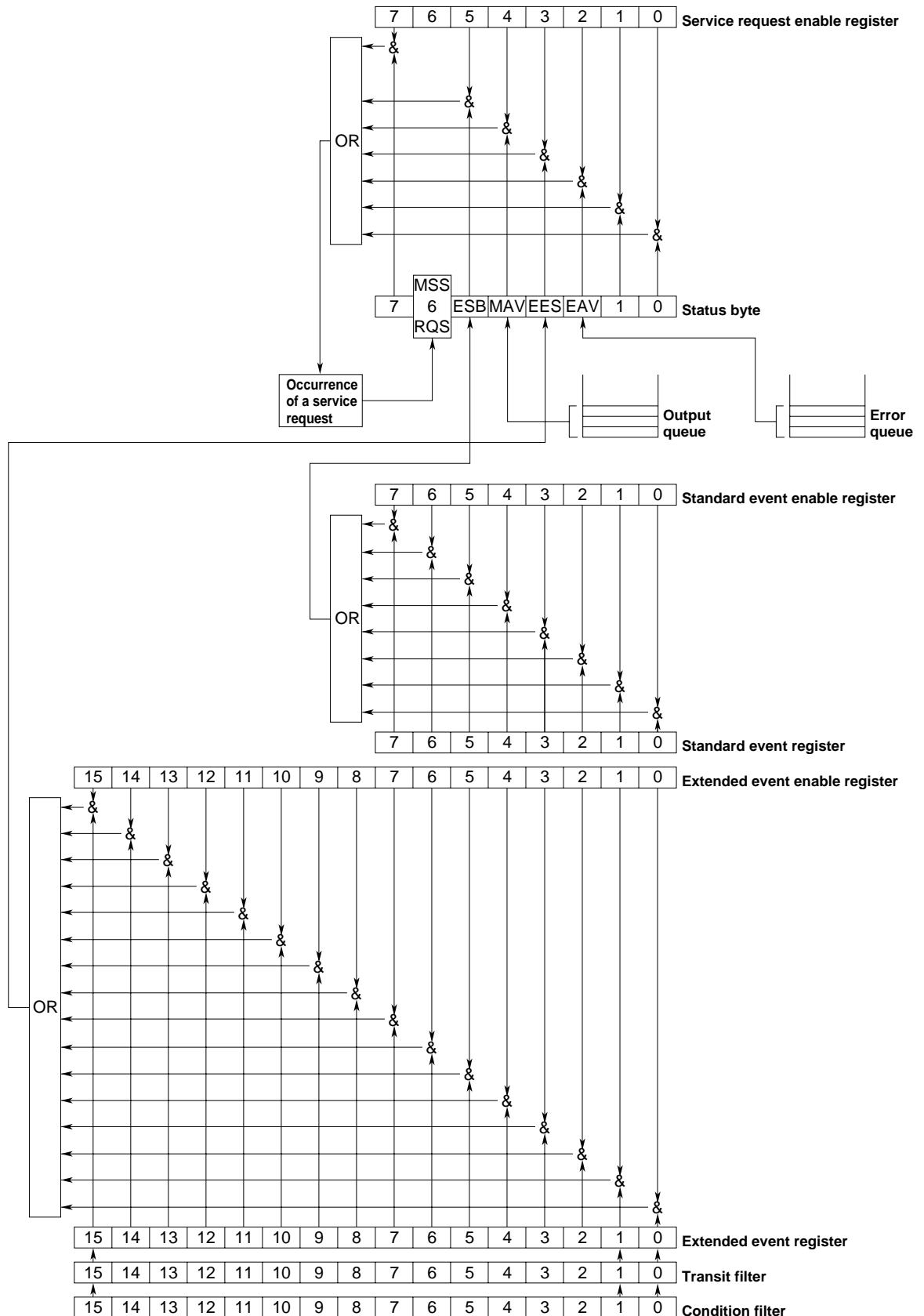
**Description** For a Description of the synchronization method using “\*WAI”, refer to page 3-7.

- Designation of overlap commands can be performed using “COMMunicate:OPSE”.

# 5 Status Report

## 5.1 Overview of the Status Report

The figure below shows the status report which is read by a serial poll. This is an extended version of the one specified in IEEE 488.2-1987.



**Overview of Registers and Queues**

Name	Function	Writing	Reading
Status byte		—	Serial poll (RQS), *STB?(MSS)
Service request enable register	Masks status byte.	*SRE	*SRE?
Standard event register	Change in device status	—	*ESR?
Standard event enable register	Masks standard event register	*ESE	*ESE?
Extended event register	Change in device status	—	STATus:EEER?
Extended event enable register	Masks standard event register	STATus:EESE	STATus:EESE?
Condition register	Current instrument status	—	STATus:CONDition?
Transit filter	Extended event occurrence conditions	STATus:FILTer <x>	STATus:FILTer <x>
Output queue	Stores response message	All executable queues to a query.	
Error queue	Stores error Nos. and messages.	—	STATus:ERRor?

**Registers and Queues which Affect the Status Byte**

Registers which affect each bit of the status byte are shown below.

- Standard event register : Sets bit 5 (ESB) of status byte to “1” or “0”.
- Output queue : Sets bit 4 (MAV) of status byte to “1” or “0”.
- Extended event register : Sets bit 3 (EES) of status byte to “1” or “0”.
- Error queue : Sets bit 2 (EAV) of status byte to “1” or “0”.

**Enable Registers**

Registers which mask a bit so that the bit does not affect the status byte, even if the bit is set to “1”, are shown below.

- Status byte : Masks bits using the service request enable register.
- Standard event register : Masks bits using the standard event enable register.
- Extended event register : Masks bits using the extended event enable register.

**Writing/Reading from Registers**

The \*ESE command is used to set bits in the standard event enable register to “1” or “0”, and the \*ESR? query is used to check whether bits in that register are set to “1” or “0”. For details of these commands, refer to Chapter 4.

**5.2 Status Byte**

**Overview of Status Byte**



**Bits 0, 1 and 7**

Not used (always “0”)

**Bit 2 EAV (Error Available)**

Set to “1” when the error queue is not empty, i.e. when an error occurs. For details, refer to page 5-5.

**Bit 3 EES (Extended Event Summary Bit)**

Set to “1” when a logical AND of the extended event register and the corresponding enable register is “1”, i.e. when an event takes place in the instrument. Refer to page 5-4.

**Bit 4 MAV (Message Available)**

Set to “1” when the output queue is not empty, i.e. when there is data which is to be output when an inquiry is made. Refer to page 5-5.

**Bit 5 ESB (Event Summary Bit)**

Set to “1” when a logical AND of the standard event register and the corresponding enable register is “1”, i.e. when an event takes place in the instrument. Refer to page 5-3.

**Bit 6 RQS (Request Status)/MSS (Master Summary Status )**

MSS is set to “1” when a logical AND of the status byte (except for bit 6) and the service request enable register is not “0”, i.e. when the instrument is requesting service from the controller.

RQS is set to “1” when MSS changes from “0” to “1”, and is cleared when a serial poll is performed or when MSS changes to “0”.

**Bit Masking**

To mask a bit in the status byte so that it does not cause an SRQ, set the corresponding bit of the service request enable register to “0”.

For example, to mask bit 2 (EAV) so that no service will be requested, even if an error occurs, set bit 2 of the service request enable register to “0”. This can be done using the \*SRE command. To query whether each bit of the service request enable register is “1” or “0”, use \*SRE?. For details of the \*SRE command, refer to Chapter 4.

**Operation of the Status Byte**

A service request is issued when bit 6 of the status byte becomes “1”. Bit 6 becomes “1” when any of the other bits becomes “1” (or when the corresponding bit in the service request enable register becomes “1”). For example, if an event takes place and the logical OR of each bit of the standard event register and the corresponding bit in the enable register is “1”, bit 5 (ESB) will be set to “1”. In this case, if bit 5 of the service request enable register is “1”, bit 6 (MSS) will be set to “1”, thus requesting service from the controller.

It is also possible to check what type of event has occurred by reading the contents of the status byte.

**Reading from the Status Byte**

The following two methods are provided for reading the status byte.

- **Inquiry using the \*STB? query**

Making an inquiry using the \*STB? query sets bit 6 to MSS. This causes the MSS to be read. After completion of the read-out, none of the bits in the status byte will be cleared.

- **Serial poll**

Execution of a serial poll changes bit 6 to RQS. This causes RQS to be read. After completion of the read-out, only RQS is cleared. Using a serial poll, it is not possible to read MSS.

**Clearing the Status Byte**

No method is provided for forcibly clearing all the bits in the status byte. Bits which are cleared are shown below.

- **When an inquiry is made using the \*STB? query**

No bit is cleared.

- **When a serial poll is performed**

Only the RQS bit is cleared.

- **When the \*CLS command is received**

When the \*CLS command is received, the status byte itself is not cleared, but the contents of the standard event register (which affects the bits in the status byte) are cleared. As a result, the corresponding bits in the status byte are cleared, except bit 4 (MAV), since the output queue cannot be emptied by the \*CLS command. However, the output queue will also be cleared if the \*CLS command is received just after a program message terminator.

**5.3 Standard Event Register****Overview of the Standard Event Register**

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

**Bit 7 PON (Power ON)**

Bit 7 PON (Power ON) Set to “1” when power is turned ON

**Bit 6 URQ (User Request)**

Not used (always “0”)

**Bit 5 CME (Command Error)**

Set to “1” when the command syntax is incorrect.

Examples: Incorrectly spelled command name; “9” used in octal data.

**Bit 4 EXE (Execution Error)**

Set to “1” when the command syntax is correct but the command cannot be executed in the current state.

Examples: Parameters are outside the setting range; an attempt is made to make a hard copy during acquisition.

**Bit 3 DDE (Device Dependent Error)**

Set to “1” when execution of the command is not possible due to an internal problem in the instrument that is not a command error or an execution error.

Example: The circuit breaker is reset.

**Bit 2 QYE (Query Error)**

Set to “1” if the output queue is empty or if the data is missing even after a query has been sent.

Examples: No response data; data is lost due to an overflow in the output queue.

**Bit 1 RQC (Request Control)**

Not used (always “0”)

**Bit 0 OPC (Operation Complete)**

Set to “1” when the operation designated by the \*OPC command has been completed. Refer to Chapter 4.

**Bit Masking**

To mask a bit in the standard event register so that it does not cause bit 5 (ESB) of the status byte to change, set the corresponding bit in the standard event enable register to “0”.

For example, to mask bit 2 (QYE) so that ESB will not be set to “1”, even if a query error occurs, set bit 2 of the standard event enable register to “0”. This can be done using the \*ESE command. To inquire whether each bit of the standard event enable register is “1” or “0”, use the \*ESE?. For details of the \*ESE command, refer to Chapter 4.



## 5.3 Standard Event Register/5.4 Extended Event Register

### Operation of the Standard Event Register

The standard event register is provided for eight different kinds of event which can occur inside the instrument. Bit 5 (ESB) of the status byte is set to "1" when any of the bits in this register becomes "1" (or when the corresponding bit of the standard event enable register becomes "1").

#### Examples

1. A query error occurs.
2. Bit 2 (QYE) is set to "1".
3. Bit 5 (ESB) of the status byte is set to "1" if bit 2 of the standard event enable register is "1".

It is also possible to check what type of event has occurred inside the instrument by reading the contents of the standard event register.

### Reading from the Standard Event Register

The contents of the standard event register can be read by the \*ESR command. After completion of the read-out, the register will be cleared.

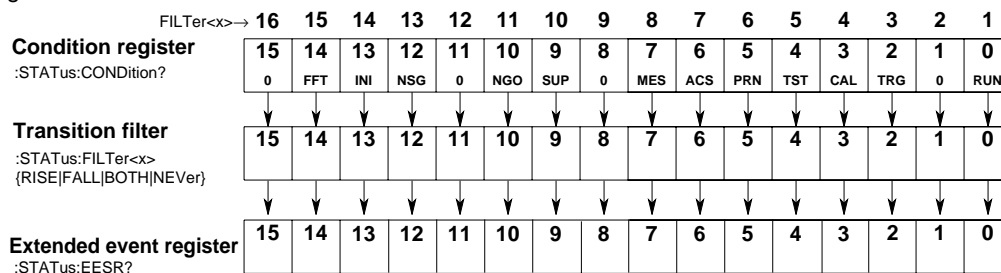
### Clearing the Standard Event Register

The standard event register is cleared in the following three cases.

- When the contents of the standard event register are read using \*ESR?
- When the \*CLS command is received
- When power is turned ON again

## 5.4 Extended Event Register

Reading the extended event register tells you whether changes in the condition register (reflecting internal conditions) have occurred. A filter can be applied which allows you to decide which events are reported to the extended event register.



The meaning of each bit of the condition register is as follows.

Bit 0 RUN (Running)	Set to "1" during acquisition.
Bit 2 TRG (Awaiting trigger)	Set to "1" when the unit is awaiting a trigger.
Bit 3 CAL (Calibrating)	Set to "1" during calibration.
Bit 4 TST (Testing)	Set to "1" during self-test.
Bit 5 PRN (Printing)	Set to "1" while the built-in printer is in operation.
Bit 6 ACS (Accessing)	Set to "1" during access of the floppy disk, hard disk, or MO disk.
Bit 7 MES (Measuring)	Set to "1" during automated measurement.
Bit 9 SUP (Set-up)	Set to "1" during auto set-up.
Bit 10 NGO (GO/No-go)	Set to "1" during GO/NO-GO.
Bit 12 NSG (N-Single)	Set to "1" during continuous acquisition when the trigger mode is single (N).
Bit 13 INI (Initializing)	Set to "1" during initialization.
Bit 14 FFT (FFT)	Set to "1" during FFT computation.

The filter is applied to each bit of the condition register separately, and can be selected from the following. Note that the numbering of the bits used in the filter setting differs from the actual bit number (1 to 16 vs. 0 to 15).

Rise	The bit of the extended event register becomes "1" when the bit of the condition register changes from "0" to "1".
Fall	The bit of the extended event register becomes "1" when the bit of the condition register changes from "1" to "0".
Both	The bit of the extended event register becomes "1" when the bit of the condition register changes from "0" to "1", or from "1" to "0".
Never	The bit of the extended event register is disabled and always "0".

## 5.5 Output Queue and Error Queue

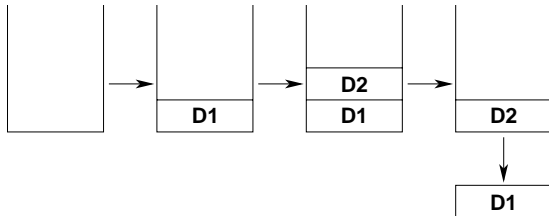
### Overview of the Output Queue

The output queue is provided to store response messages to queries. For example, when the `WAVEform:SEND?` query is sent to request output of the acquired waveform, the response data will be stored in the output queue until it is read out.

The example below shows that data is stored record by record in the output queue, and is read out oldest item first, newest item last. The output queue is emptied in the following cases (in addition to when read-out is performed).

- When a new message is received from the controller
- When dead lock occurs (page 3-2)
- When a device clear command (DCL or SDC) is received
- When power is turned ON again

The output queue cannot be emptied using the `*CLS` command. To see whether the output queue is empty or not, check bit 4 (MAV) of the status byte.



### Overview of the Error Queue

The error queue stores the error No. and message when an error occurs. For example, when the built-in battery has run out, an error occurs and its error No. (901) and message "Backup Failure" will be stored in the error queue.

The contents of the error queue can be read using the `STATUS:ERROR?` query. As with the output queue, messages are read oldest first, newest last (refer to the previous page).

If the error queue becomes full, the final message will be replaced by message 350, "Queue overflow".

The error queue is emptied in the following cases (in addition to when read-out is performed).

- When the `*CLS` command is received
- When power is turned ON again

To see whether the error queue is empty or not, check bit 2 (EAV) of the status byte.

## 6 Sample Program

This section describes sample programs for a IBM PC/AT and compatible system with National Instruments AT-GPIB/TNT IEEE-488.2 board installed. Sample programs in this manual are written in Quick BASIC version 4.0/4.5.

### 6.1 Before Programming

#### Setting up

##### Initialization

None of the sample programs given in this chapter include initialization, so be sure to initialize the instrument using the INITIALIZE key before running the programs.

##### Address 1

All the sample programs given in this chapter use address 1 for this instrument, so be sure to assign the instrument to address 1 as described on page 1-4.

#### Setting up the Personal Computer

##### Terminator for binary transmission

When outputting waveform data in binary code, make sure that the terminator for this instrument is set to EOI on the personal computer.

##### Input buffer length for binary transmission

Be careful when receiving BINARY data that the received data does not overrun the capacity of the receive buffer in the personal computer, which may be as small as 255 bytes in some cases.

## 6.2 Setting the T/div

### Sample Program

```

*****
*
* DL1540C Sample Program1 for GPIB interface
*                               Microsoft QuickBASIC 4.0/4.5 Version
*
*****
*
* DL1540C T/div Set & Get
*
*****
'
'
'   REM $INCLUDE: 'qbdecl4.bas'
'
'   DEVNAME$ = "dl1500"
'   CALL IBFIND(DEVNAME$, DEV%)
'
'   CMD$ = "COMMUNICATE:HEADER OFF"
'   CALL IBWRT(DEV%, CMD$)
'
'   CMD$ = "TIMEBASE:TDIV 2US"
'   CALL IBWRT(DEV%, CMD$)
'
'   dt$ = SPACE$(200)
'   QRY$ = "TIMEBASE:TDIV?"
'   CALL IBWRT(DEV%, QRY$)
'   CALL IBRD(DEV%, dt$)
'
'   PRINT "TIME/DIV = "; dt$
'
'
END

*****
*
* DL1540C Sample Program1 for RS-232C
*                               Microsoft QuickBASIC 4.0/4.5 Version
*
*                               Rate:9600 Parity:None CHR:8 STOPBIT:1 XON/XON Term:CR+LF
*
*****
*
* DL1540C T/div Set & Get
*
*****
'
'   OPEN "COM1:9600,N,8,1,ASC,CS0,DS0,LF" FOR RANDOM AS #1
'
'
'   PRINT #1, "COMMUNICATE:HEADER OFF"
'
'   PRINT #1, "TIMEBASE:TDIV 2US"           'Set T/div 2us
'
'   dt$ = SPACE$(200)
'   PRINT #1, "TIMEBASE:TDIV?"           'Get T/div
'   LINE INPUT #1, dt$
'
'   PRINT "TIME/DIV = "; dt$
'
'
'   CLOSE #1
'
END

```

## 6.3 Reading the Latest Waveform Data of CH1 (Ascii Data)

### Sample Program

```

*****
*
* DL1540C Sample Program3 for GPIB interface
*                               Microsoft QuickBASIC 4.0/4.5 Version
*
*****
*
* Read latest CH1 waveform data (use ASCII format)
*
*****
'
'
REM $INCLUDE: 'qbdecl4.bas'
'
DEVICE$ = "DL1500": CALL IBFIND(DEVICE$, DL%)
                    CALL IBSIC(DL%)
BORD$ = "GPIB0": CALL IBFIND(BORD$, BD%)
                    CALL IBSIC(BD%)
V% = 1: CALL IBSRE(BD%, V%)
                    CALL IBCLR(DL%)
'
' Setup waveform send condition
CMD$ = "STOP": CALL IBWRT(DL%, CMD$)
CMD$ = "WAVEFORM:TRACE 1;DATASELECT ACQDATA;FORMAT ASCII;BYTEORDER LSBFIRST;START 0;END
10019"
                    CALL IBWRT(DL%, CMD$)
CMD$ = "COMMUNICATE:HEADER OFF"
                    CALL IBWRT(DL%, CMD$)
' Get Data Length
CMD$ = "WAVEFORM:LENGTH?"
                    CALL IBWRT(DL%, CMD$)
V$ = SPACE$(100): CALL IBRD(DL%, V$)
V% = INSTR(V$, CHR$(10))
L% = VAL(LEFT$(V$, V% - 1))
M% = L% MOD 10
C% = (L% - M%) / 10
C% = C% * 10
' Get waveform data
D% = 0
V$ = SPACE$(210)
' Get 10 datas at once
IF C% = 0 THEN END
FOR I% = 0 TO C% - 1 STEP 10
    CMD$ = "WAVEFORM:START" + STR$(I%) + ";END" + STR$(I% + 9) + ";SEND?"
                    CALL IBWRT(DL%, CMD$)
                    CALL IBRD(DL%, V$)

    K% = 1
    FOR J% = 0 TO 9
        IF J% < 9 THEN S% = INSTR(K%, V$, ",") ELSE S% = INSTR(K%, V$, CHR$(10))
        PRINT VAL(MID$(V$, K%, S% - K%))
        K% = S% + 1
    NEXT J%
NEXT I%
CMD$ = "WAVEFORM:START" + STR$(C%) + ";END" + STR$(C% + M% - 1) + ";SEND?"
                    CALL IBWRT(DL%, CMD$)
                    CALL IBRD(DL%, V$)

K% = 1
FOR J% = 0 TO M% - 1
    IF J% < M% - 1 THEN S% = INSTR(K%, V$, ",") ELSE S% = INSTR(K%, V$, CHR$(10))
    PRINT VAL(MID$(V$, K%, S% - K%))
    K% = S% + 1
NEXT J%
END

```

### 6.3 Reading the Latest Waveform Data of CH1 (Ascii Data)

---

```
*****
*
* DL1540C Sample Program2 for RS-232C
* Microsoft QuickBASIC 4.0/4.5 Version
*
* Rate:9600 Parity:None CHR:8 STOPBIT:1 XON/XON Term:CR+LF
*
*****
*
* Read latest CH1 waveform data (use ascii format)
*
*****
'
OPEN "COM1:9600,N,8,1,ASC,CS0,DS0,LF" FOR RANDOM AS #1
'
' Setup waveform send condition
PRINT #1, "STOP"
PRINT #1, "WAVEFORM:TRACE 1;DATASELECT ACQDATA;FORMAT ASCII;START 0;END 10031"
PRINT #1, "COMMUNICATE:HEADER OFF"
' Get Data Length
PRINT #1, "WAVEFORM:LENGTH?"
LINE INPUT #1, V$
L% = VAL(V$)
IF L% = 0 THEN END
' Get waveform data
PRINT #1, "WAVEFORM:SEND?"
FOR I% = 1 TO L%
    INPUT #1, D$
    PRINT I%, D$
NEXT I%
PRINT #1, "COMMUNICATE:HEADER ON"
PRINT #1, "COMMUNICATE:STATUS?"
LINE INPUT #1, V$
PRINT V$
CLOSE #1
END
```

## 6.4 Reading the Latest Waveform Data of CH1 (Binary Data)

### Sample Program

```

*****
*
* DL1540C Sample Program2 for GPIB interface
*                               Microsoft QuickBASIC 4.0/4.5 Version
*
*****
*
* Read latest CH1 waveform data (use binary format)
*
*****
'
'
REM $INCLUDE: 'qbdecl4.bas'
'
DEVICE$ = "DL1500": CALL IBFIND(DEVICE$, DL%)
                    CALL IBSIC(DL%)
BORD$ = "GPIB0": CALL IBFIND(BORD$, BD%)
                    CALL IBSIC(BD%)
V% = 1: CALL IBSRE(BD%, V%)
                    CALL IBCLR(DL%)
'
' Setup waveform send condition
CMD$ = "STOP": CALL IBWRT(DL%, CMD$)
CMD$ = "WAVEFORM:TRACE 1;DATASELECT ACQDATA;FORMAT WORD;BYTEORDER LSBFIRST;START 0;END
10019"
                    CALL IBWRT(DL%, CMD$)
' Get V/div and offset, these are needed to convert binary to voltage
CMD$ = "COMMUNICATE:HEADER OFF;:WAVEFORM:VDIV?;OFFSET?"
                    CALL IBWRT(DL%, CMD$)
V$ = SPACE$(100): CALL IBRD(DL%, V$)
V% = INSTR(V$, ";")
V! = VAL(LEFT$(V$, V% - 1))
O! = VAL(MID$(V$, V% + 1, INSTR(V$, CHR$(10)) - V% - 1))
' Get Data Length
CMD$ = "WAVEFORM:LENGTH?"
                    CALL IBWRT(DL%, CMD$)
V$ = SPACE$(100): CALL IBRD(DL%, V$)
V% = INSTR(V$, CHR$(10))
L% = VAL(LEFT$(V$, V% - 1))
M% = L% MOD 100
C% = (L% - M%) / 100
C% = C% * 100
' Get waveform data
V$ = SPACE$(210)
' Get 100 datas at once
IF C% = 0 THEN END
FOR I% = 0 TO C% - 1 STEP 100
    CMD$ = "WAVEFORM:START" + STR$(I%) + ";END" + STR$(I% + 99) + ";SEND?"
                    CALL IBWRT(DL%, CMD$)
                    CALL IBRD(DL%, V$)

    FOR J% = 0 TO 99
        de = CVI(MID$(V$, J% * 2 + 9, 2))
        PRINT (CVI(MID$(V$, J% * 2 + 9, 2)) AND &HFFFE) * V! / 3200 + O!
    NEXT J%
NEXT I%
CMD$ = "WAVEFORM:START" + STR$(C%) + ";END" + STR$(C% + M% - 1) + ";SEND?"
                    CALL IBWRT(DL%, CMD$)
                    CALL IBRD(DL%, V$)

FOR J% = 0 TO M% - 1
    PRINT (CVI(MID$(V$, J% * 2 + 9, 2)) AND &HFFFE) * V! / 3200 + O!
NEXT J%
END

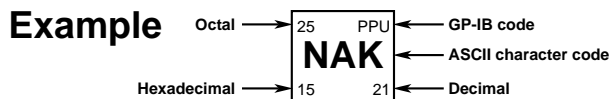
```

# Appendix

## Appendix 1 ASCII Character Codes

ASCII character codes are given below.

	0	1	2	3	4	5	6	7
0	<sup>0</sup> NUL	<sup>20</sup> DEL	<sup>40</sup> SP	<sup>0 60</sup> 0	<sup>100</sup> @	<sup>0 120</sup> P	<sup>140</sup> '	<sup>0 160</sup> p
1	<sup>1</sup> SOH	<sup>21</sup> DC1	<sup>41</sup> !	<sup>1 61</sup> 1	<sup>101</sup> A	<sup>1 121</sup> Q	<sup>141</sup> a	<sup>1 161</sup> q
2	<sup>2</sup> STX	<sup>22</sup> DC2	<sup>42</sup> "	<sup>2 62</sup> 2	<sup>102</sup> B	<sup>2 122</sup> R	<sup>142</sup> b	<sup>2 162</sup> r
3	<sup>3</sup> ETX	<sup>23</sup> DC3	<sup>43</sup> #	<sup>3 63</sup> 3	<sup>103</sup> C	<sup>3 123</sup> S	<sup>143</sup> c	<sup>3 163</sup> s
4	<sup>4</sup> EOT	<sup>24</sup> DC4	<sup>44</sup> \$	<sup>4 64</sup> 4	<sup>104</sup> D	<sup>4 124</sup> T	<sup>144</sup> d	<sup>4 164</sup> t
5	<sup>5</sup> ENQ	<sup>25</sup> NAK	<sup>45</sup> %	<sup>5 65</sup> 5	<sup>105</sup> E	<sup>5 125</sup> U	<sup>145</sup> e	<sup>5 165</sup> u
6	<sup>6</sup> ACK	<sup>26</sup> SYN	<sup>46</sup> &	<sup>6 66</sup> 6	<sup>106</sup> F	<sup>6 126</sup> V	<sup>146</sup> f	<sup>6 166</sup> v
7	<sup>7</sup> BEL	<sup>27</sup> ETB	<sup>47</sup> ,	<sup>7 67</sup> 7	<sup>107</sup> G	<sup>7 127</sup> W	<sup>147</sup> g	<sup>7 167</sup> w
8	<sup>10</sup> BS	<sup>30</sup> CAN	<sup>50</sup> (	<sup>8 70</sup> 8	<sup>110</sup> H	<sup>8 130</sup> X	<sup>150</sup> h	<sup>8 170</sup> x
9	<sup>11</sup> HT	<sup>31</sup> EM	<sup>51</sup> )	<sup>9 71</sup> 9	<sup>111</sup> I	<sup>9 131</sup> Y	<sup>151</sup> i	<sup>9 171</sup> y
A	<sup>12</sup> LF	<sup>32</sup> SUB	<sup>52</sup> *	<sup>10 72</sup> :	<sup>112</sup> J	<sup>10 132</sup> Z	<sup>152</sup> j	<sup>10 172</sup> z
B	<sup>13</sup> VT	<sup>33</sup> ESC	<sup>53</sup> +	<sup>11 73</sup> ;	<sup>113</sup> K	<sup>11 133</sup> [	<sup>153</sup> k	<sup>11 173</sup> {
C	<sup>14</sup> FF	<sup>34</sup> FS	<sup>54</sup> ,	<sup>12 74</sup> <	<sup>114</sup> L	<sup>12 134</sup> \	<sup>154</sup> l	<sup>12 174</sup> 
D	<sup>15</sup> CR	<sup>35</sup> GS	<sup>55</sup> -	<sup>13 75</sup> =	<sup>115</sup> M	<sup>13 135</sup> ]	<sup>155</sup> m	<sup>13 175</sup> }
E	<sup>16</sup> SO	<sup>36</sup> RS	<sup>56</sup> .	<sup>14 76</sup> >	<sup>116</sup> N	<sup>14 136</sup> ^	<sup>156</sup> n	<sup>14 176</sup> ~
F	<sup>17</sup> SI	<sup>37</sup> US	<sup>57</sup> /	<sup>15 77</sup> ?	<sup>UNL</sup> O	<sup>15 137</sup> _	<sup>157</sup> o	<sup>15 177</sup> DEL (RUBOUT)
	Address Command	Universal Command	Listener Address	Talker Address	Secondary Command			



App  
Appendix



## Appendix 2 Error Messages

Error messages related to communications are given below.

When servicing is required, contact your nearest YOKOGAWA representative, as given on the back cover of this manual.

Only error messages relating to communications are given here. For other error messages, refer to the User's Manual IM 701530-01E.

### Errors in communications commands (100 to 199)

Code	Message	Action	Reference Page
102	Syntax error	Incorrect syntax	Chapter 3, Chapter 4
103	Invalid separator	Insert a comma between data items to separate them.	3-1
104	Data type error	Refer to pages 2-8 to 2-11 and enter data using the correct data format.	3-5 to 3-6
105	GET not allowed	GET is not supported as a response to an interface message.	1-5
108	Parameter not allowed	Check the number of parameters.	3-5, Chapter 4
109	Missing parameter	Enter the required number of parameters.	3-5, Chapter 4
111	Header separator error	Insert a space between the header and the data to separate them.	3-2
112	Program mnemonic too long	Check the mnemonic (character string consisting of letters and numbers).	Chapter 4
113	Undefined header	Check the header.	Chapter 4
114	Header suffix out of range	Check the header.	Chapter 4
120	Numeric data error	Mantissa must be entered before the numeric value in <NRf> format.	3-5
123	Exponent too large	Use a smaller exponent in <NR3> format.	3-5, Chapter 4
124	Too many digits	Limit the number of digits to 255 or less.	3-5, Chapter 4
128	Numeric data not allowed	Enter in a format other than <NRf> format.	3-5, Chapter 4
131	Invalid suffix	Check the units for <Voltage>, <Time> and <Frequency>.	3-5
134	Suffix too long	Check the units for <Voltage>, <Time> and <Frequency>.	3-5
138	Suffix not allowed	No units are allowed other than <Voltage>, <Time> and <Frequency>.	3-5
141	Invalid character data	Enter one of the character strings in {... ... ...}.	Chapter 4
144	Character data too long	Check the character strings in {... ... ...}.	Chapter 4
148	Character data not allowed	Enter in a format other than one of those in {... ... ...}.	Chapter 4
150	String data error	<Character string> must be enclosed by double quotation marks or single quotation marks.	3-6
151	Invalid string data	<Character string> is too long or contains characters which cannot be used.	Chapter 4
158	String data not allowed	Enter in a data format other than <Character string>.	Chapter 4
161	Invalid block data	<Block data> is not allowed.	3-6, Chapter 4
168	Block data not allowed	<Block data> is not allowed.	3-6, Chapter 4
171	Invalid expression	Equation is not allowed.	Chapter 4
178	Expression data not allowed	Equation is not allowed.	Chapter 4
181	Invalid outside macro definition	Does not conform to the macro definition specified in IEEE488.2.	—

**Errors in communications execution (200 to 299)**

Code	Message	Action	Reference Page
221	Setting conflict	Check the relevant setting.	Chapter 4
222	Data out of range	Check the setting range.	Chapter 4
223	Too much data	Check the data byte length.	Chapter 4
224	Illegal parameter value	Check the setting range.	Chapter 4
241	Hardware missing	Check availability of options.	—
260	Expression error	Equation is not allowed.	—
270	Macro error	Does not conform to the macro definition specified in IEEE488.2.	—
272	Macro execution error	Does not conform to the macro definition specified in IEEE488.2.	—
273	Illegal macro label	Does not conform to the macro definition specified in IEEE488.2.	—
275	Macro definition too long	Does not conform to the macro definition specified in IEEE488.2.	—
276	Macro recursion error	Does not conform to the macro function specified in IEEE488.2.	—
277	Macro redefinition not allowed	Does not conform to the macro definition specified in IEEE488.2.	—
278	Macro header not found	Does not conform to the macro definition specified in IEEE488.2.	—

**Errors in communications Queries (400 to 499)**

Codes	Message	Action	Reference Page
410	Query INTERRUPTED	Check transmission/reception order.	3-2
420	Query UNTERMINATED	Check transmission/reception order.	3-2
430	Query DEADLOCKED	Limit the length of the program message including <PMT> to 1024 bytes or less.	3-2
440	Query UNTERMINATED after indefinite response	Do not enter any query after *IDN? and *OPT?.	—

**Errors in Execution (600 to 899)**

Codes	Message	Action	Reference Page
600 to 899	Execution error	Refer to the User's Manual 701530-01E.	—
750	Send error	Data cannot be sent during data acquisition. Stop data acquisition and try again.	—
751	GO/NO-GO execute error	Verify whether the GO/NO-GO Mode is set to ON, and verify the type setting.	4-31, 4-33
752	Zone edit error	Verify the zone editing conditions	4-33 to 4-35

## Appendix 2 Error Messages

---

### Errors in System Operation (912 to 915)

Code	Message	Action	Reference Page
912	Fatal error in Communication driver	Service is required.	—
914	Time out occurs in Communication	Send the data within the time-out limit. Or change the communication board if any damage is present.	—
915	Can't detect listener	Verify if the plotter or AG are set to listener mode. Check GP-IB connector	—

### Warnings (50, 350, 390)

Code	Message	Action	Reference Page
50	*OPC/? exists in message	*OPC or *OPC? must be positioned at the end of the message.	—
350	Queue overflow	Read out the queue.	5-5
390	Overrun error (only for RS-232C)	Adjust the baud rate.	—

#### **Note**

The warning code 350 only appears in case of an overflow of the error queue. The error which occurs in case of clearing the STATus:ERRor? will not appear on the screen.

---

## Appendix 3 Overview of IEEE 488.2-1992

The GP-IB interface provided with this instrument conforms to IEEE 488.2-1992. This standard requires the following 23 points be stated in this document. This appendix describes these points.

- (1) **Subsets supported by IEEE 488.1 interface functions**  
Refer to the specifications on page 1-3.
- (2) **Operation of device when the device is assigned to an address other than one of the addresses 0 to 30**  
This instrument does not allow assignment to an address other than 0 to 30.
- (3) **Reaction when the user initializes address settings.**  
Change of the current address is acknowledged when a new address is set using the MISC key menu. The newly set address is valid until another new address is set.
- (4) **Device set-up at power ON. Commands which can be used at power ON**  
Basically, the previous settings (i.e. the settings which were valid when power was turned OFF) are valid.  
All commands are available at power ON.
- (5) **Message transmission options**
  - (a) **Input buffer size and operation**  
The input buffer's capacity is 1024 bytes.
  - (b) **Types of queries which return multiple response messages**  
Refer to the examples of each command in Chapter 4.
  - (c) **Types of queries which generate response data during analysis of the syntax**  
Every query generates response data when analysis of the syntax is performed.
  - (d) **Types of queries which generate response data during reception**  
No query generates response data when it is received by the controller.
  - (e) **Types of commands which have pairs of parameters.**  
Refer to the examples of each command in Chapter 4.
- (6) **List of function elements which configure commands used for the device. All those which are included in elements of composite command program headers**  
Refer to Chapter 3 and 4.
- (7) **Buffer size which affects transmission of block data**  
The transmission buffer's capacity is 64 K bytes.
- (8) **List of program data elements which can be used in equations and nesting limit**  
Cannot be used.
- (9) **Syntax of response data to queries**  
Refer to the examples of each command in Chapter 4..
- (10) **Communication between devices which do not follow the rules regarding response data**  
Data output to an HP-GL plotter and AG series unit in talk-only mode
- (11) **Size of data block of response data**  
0 to 240512 bytes (for DL1540C)  
0 to 4008032 bytes (for DL1540CL)
- (12) **List of supported common commands**  
Refer to 4.31 Common Command Group.
- (13) **Condition of device when calibration is successfully completed**  
While acquisition is stopped ..... acquisition data is not corrected.  
While acquisition is in progress .. acquisition data is corrected and displayed.
- (14) **Maximum length of block data which can be used for definition of trigger macro when \*DDT is used**  
\*DDT is not supported.
- (15) **Maximum length of macro label if macro definition is used; maximum length of block data which can be used for definition of macro; processing when recursion is used in definition of macro**  
Macro functions are not supported.
- (16) **Response to \*IDN?**  
Refer to 4.31 Common Command Group.
- (17) **Size of storage area for protected user data if PUD and \*PUD? are used.**  
\*PUD and \*PUD? are not supported.
- (18) **Length of resource name if \*RDT and \*RDT? are used.**  
\*RDT and \*RDT? are not supported.
- (19) **Change in status if \*RST, \*LRN?, \*RCL and \*SAV are used.**  
\*RST  
Refer to 4.31 Common Command Group.  
\*LRN?, \*RCL, \*SAV  
These commands are not supported.
- (20) **Execution range of self-test using \*TST?**  
Board test (for each internal memory) given in the Selfest selection menu displayed using the MISC key can be executed.
- (21) **Structure of extended return status**  
Refer to chapter 5.
- (22) **To find out whether each command is performed in parallel or sequentially**  
Refer to 3.5 Synchronization with the Controller, or chapter 4.
- (23) **Functions performed until a message indicating completion of the command is displayed**  
Refer to the function description of each command in chapter 4, and to the User's Manual 701530-01E.

# Index

## Symbols

^END ..... 3-1

## A

ACCumulate? ..... 4-22  
 ASCII Character Codes ..... App-1  
 Address commands ..... 1-5  
 abbreviated format/full format ..... 4-16  
 acquisition  
   mode ..... 4-11  
   start ..... 4-58  
   stop ..... 4-59  
 acquisition mode ..... 4-11  
 address, setting ..... 1-4  
 auto calibration ..... 4-12  
 auto set-up ..... 4-12  
 automatic measurement ..... 4-50

## B

Block data ..... 3-6  
 Boolean ..... 3-6  
 bandwidth limits ..... 4-28  
 battery ..... 4-61  
 bit length of the waveform data ..... 4-71

## C

Character data ..... 3-6  
 Character string ..... 3-6  
 Commands ..... 3-3  
 Common command header ..... 3-3  
 Compound header ..... 3-3  
 Concatenating Commands ..... 3-3  
 Connecting the GP-IB Cable ..... 1-2  
 Connecting the RS-232-C Interface Cable ..... 2-2  
 calibration ..... 4-12  
 channel vertical axis  
   display ON/OFF ..... 4-14  
   input conditions ..... 4-13  
   vertical position ..... 4-14  
   voltage axis sensitivity ..... 4-14  
 clear trace ..... 4-15  
 clearing registers at power ON ..... 4-75  
 command list ..... 4-1  
 common command ..... 4-74  
 computation ..... 4-46  
 condition register ..... 4-59  
 configuration function ..... 4-61  
 cursors ..... 4-18

## D

DCL (Device Clear) ..... 1-5  
 Data ..... 3-5  
 Dead lock ..... 3-2  
 date ..... 4-61  
 decimal number ..... 3-5  
 distal/mesial/proximal points ..... 4-53

## E

Enable Registers ..... 5-2  
 edge trigger ..... 4-66  
 error code and message (query) ..... 4-59  
 error message ON/OFF ..... 4-59  
 error message list ..... App-2  
 error queue ..... 4-59, 5-5  
 expansion ..... 4-73  
 extended event  
   awaiting an extended event ..... 4-16  
   extended event enable register ..... 4-59  
   extended event register ..... 4-59, 5-4  
   generating a response to an extended event ..... 4-16  
   register ..... 3-8

## F

Frequency ..... 3-5  
 Front Panel ..... 1-1, 2-1  
 filter ..... 4-28  
 floppy disk  
   deleting ..... 4-25  
   formatting ..... 4-26  
   loading ..... 4-26  
   protecting ..... 4-27  
   saving ..... 4-27

## G

GET (Group Execute Trigger) ..... 1-5  
 GP-IB Interface Specifications ..... 1-3  
 GP-IB connector ..... 1-2  
 GTL (Go To Local) ..... 1-5  
 graticule (scale) ..... 4-23  
 GO/NO-GO determination  
   determination ON/OFF ..... 4-31  
   parameter determination ..... 4-31  
   zone determination ..... 4-33

## H

HF rejection ..... 4-69  
 Header interpretation rules ..... 3-4  
 header ON/OFF ..... 4-15  
 hold-off time ..... 4-66

## I

IDY (Identify) ..... 1-5  
 IFC (Interface Clear) ..... 1-5  
 initialization ..... 4-45, 4-75  
 input coupling ..... 4-14  
 inquiry about calibration result ..... 4-74  
 inquiry about the model ..... 4-75  
 intensity ..... 4-23  
 internal memory  
   recalling ..... 4-55  
   storing ..... 4-60  
 interpolation method ..... 4-23

## Index

<b>L</b>	<b>Page</b>
LLO (Local Lockout) .....	1-5
LOCAL key .....	1-2
Listener function .....	1-2
lithium battery .....	4-61

<b>M</b>	<b>Page</b>
MISC key .....	1-4
Messages .....	3-1
Mnemonic .....	3-3
Multi-line messages .....	1-5
Multiplier .....	3-5

<b>N</b>	<b>Page</b>
NL .....	3-1
NL^END .....	3-1
NR form .....	3-5
Name and Function of Each Part .....	1-1, 2-1

<b>O</b>	<b>Page</b>
OR trigger .....	4-67
Overlap	
overlap commands .....	3-7
Overview of IEEE 488.2 .....	App-5
offset voltage .....	4-14, 4-72
operation pending register .....	4-16
output of screen data	
HP-GL plotter .....	4-41
output equipment .....	4-39
printer .....	4-42
output queue .....	5-5
output to a plotter .....	4-41
output to a printer .....	4-40
outputting waveform data in ASCII format .....	4-72
outputting waveform data in byte format .....	4-72
outputting waveform data in word format .....	4-72
overlap	
awaiting the command following "*WAI" until end of ..	4-76
command used for overlap .....	4-16
generating response at end of overlap .....	4-75
overlap enable register .....	4-16
setting OPC event at end of overlap command .....	4-75

<b>P</b>	<b>Page</b>
PMT .....	3-1
PPC (Parallel Poll Configure) .....	1-5
PPU (Parallel Poll Unconfigure) .....	1-5
Program data .....	3-1
Program header .....	3-1
Program messages .....	3-1
pattern trigger .....	4-67
phase shift .....	4-47
printer .....	4-40
printing density .....	4-61
probe attenuation .....	4-14

<b>Q</b>	<b>Page</b>
Query .....	3-2, 3-4
query	
abbreviated or full response method .....	4-16
querying installed options .....	4-75

<b>R</b>	<b>Page</b>
REMOTE indicator .....	1-2
REN (Remote Enable) .....	1-5
RMT .....	3-1
Rear Panel .....	1-1, 2-1
Register .....	3-5
Response .....	3-4
Response data .....	3-2
Response header .....	3-2
Response messages .....	3-1
Responses to Interface Messages .....	1-5
recalling (from internal memory) .....	4-55
record length .....	4-11
repetitive sampling mode ON/OFF .....	4-11
response	
abbreviated of full response .....	4-16
response header ON/OFF .....	4-15

<b>S</b>	<b>Page</b>
SDC (Selected Device Clear) .....	1-5
SPD (Serial Poll Disable) .....	1-5
SPE (Serial Poll Enable) .....	1-5
Sample Program .....	6-1
Sequential command .....	3-7
Simple header .....	3-3
Status Byte .....	5-2
Switching from local to remote mode .....	1-2
Switching from remote to local mode .....	1-2
sample rate .....	4-62, 4-72
scale (graticule) .....	4-23
scale ON/OFF .....	4-23
service request enable register .....	4-75, 5-3
setting the instrument .....	1-4, 2-7
simple trigger .....	4-66
smoothing ON/OFF .....	4-28
snapshot .....	4-58
standard event enable register .....	4-74, 5-3
standard event register .....	4-75, 5-4
start acquisition .....	4-58
starting (acquisition) .....	4-58
status byte .....	4-76
status function .....	4-59
status report .....	5-1
stop acquisition .....	4-59
stopping (acquisition) .....	4-59
storing (internal memory) .....	4-60
synchronization using "STATus:CONDition?" .....	3-8
synchronization with the controller .....	3-7
system .....	4-61

<b>T</b>	<b>Page</b>
T/div .....	4-62
TCT (Take Control) .....	1-5
TV trigger .....	4-67
Talk-only function .....	1-2
Talker function .....	1-2
time .....	4-62
time axis expansion .....	4-73
time axis setting .....	4-62
timebase .....	4-62
transition filter .....	4-59
trigger .....	4-65

<b>U</b>	<b>Page</b>
Uni-line messages .....	1-5
Universal commands .....	1-5
Upper-level query .....	3-4

<b>V</b>	<b>Page</b>
V/div .....	4-14
Voltage .....	3-5
variable ON/OFF .....	4-14
vertical position .....	4-14
voltage axis sensitivity .....	4-14

<b>W</b>	<b>Page</b>
waveform data .....	4-71
waveform expansion .....	4-73
waveform information ON/OFF .....	4-23
waveform inversion ON/OFF .....	4-14
wide screen ON/OFF .....	4-23
width trigger .....	4-68

<b>X</b>	<b>Page</b>
X-Y waveform display ON/OFF .....	4-23

<b>Z</b>	<b>Page</b>
zoom .....	4-73