DL1540C/1540CL

Digital Oscilloscope
Communication Interface

USER'S MANUAL

YOKOGAWA
Yokogawa Electric Corporation

IM 701530-11E
2nd Edition

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
2nd Edition: May 2000

Disk No. DL26
2nd Edition: May 2000 (YK)
All Right Reserved, Copyright © 1998 Yokogawa Electric Corporation

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
Sample programs, written in Visual BASIC, for MS-DOS/V machines equipped with the following GP-IB board: AT-GPIB/TNT IEEE-488.2, from National Instruments.

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	***Note***	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.

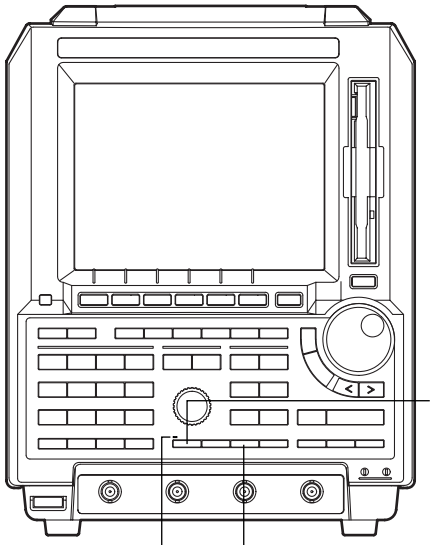
Symbol	Description	Example	Example
< >	Defined value	CHANnel<x> <x>=1, 2	CHANNEL2
{}	One of the options in {} is selected.	MODE {AND|OR}	MODE AND
|	Exclusive OR	MODE {AND|OR}	MODE AND
[ ]	Abbreviated	:MEASure[:MODE] {<NRf>}	
...	may be repeated		

Contents

1 Overview of the GP-IB Function

1.1 Name and Function of Each Part

Front Panel

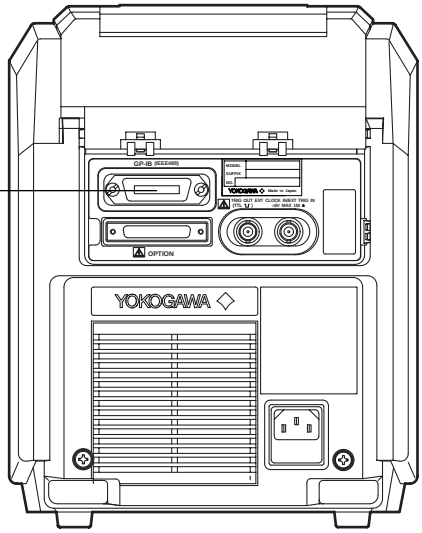


LOCAL key
Press this key to switch from remote mode to local mode so as to enable key operation.

REMOTE indicator
Lit while remote mode is active via communications.

MISC key
Press this key to set communication settings.

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



GP-IB connector
Used to connect a controller (personal computer etc.) or plotter using a GP-IB cable.

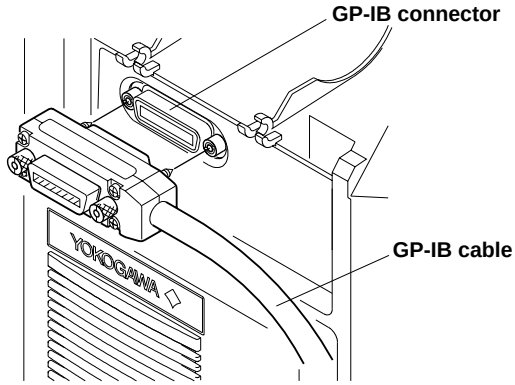
1.2 Connecting the GP-IB Cable

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.

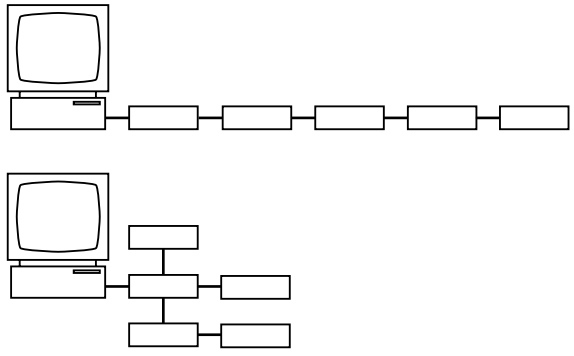
Connection Method

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.

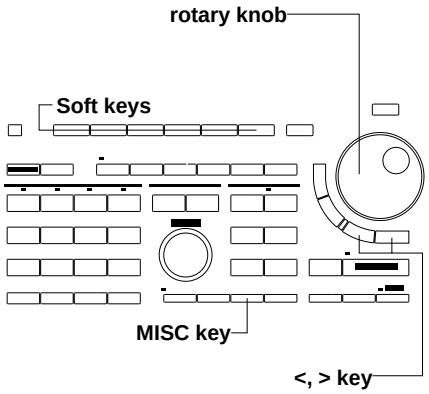
Remote mode clear : Remote mode can be cleared by pressing the LOCAL key. However, this is not possible if Local Lockout has been set by the controller.

Mechanical Specifications

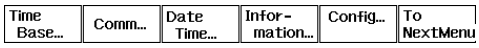
Function	Subset Name	Description
Source handshaking	SH1	Full source handshaking capability
Acceptor handshaking	AH1	Full acceptor handshaking capability
Talker	T5	Basic talker capability, serial polling, untalk on MLA(My Listen Address), talk-only capability
Listener	L4	Basic listener capability, unlisten 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.



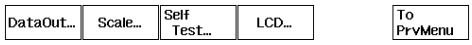
3. Press the "**GP-IB**" soft key.

Setting the Address

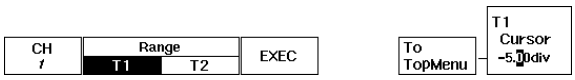
4. Turn the rotary knob to set the desired address.

Selecting Data Output (Talk-only Mode)

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



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



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.

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.

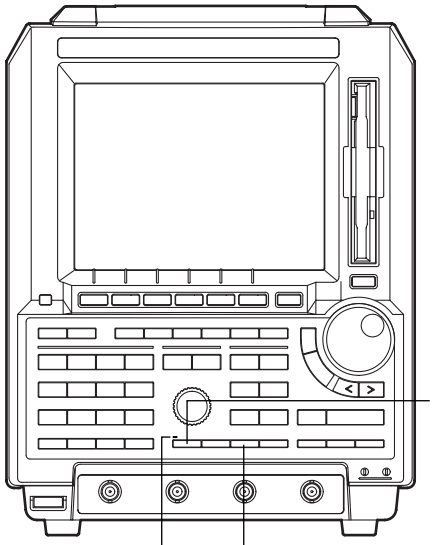
Responses to a multi-line message (universal command)

- 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

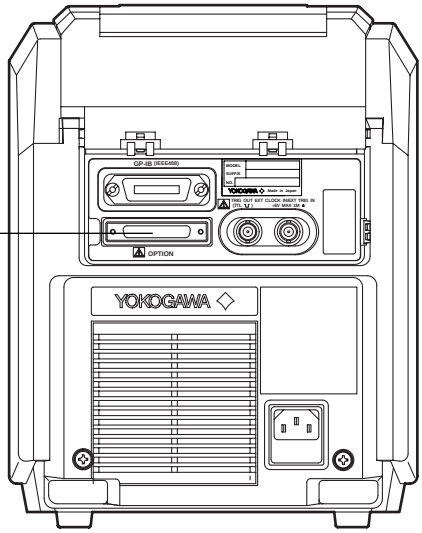


LOCAL key
Press this key to switch from remote mode to local mode so as to enable key operation.

REMOTE indicator
Lit while remote mode is active via communications.

MISC key
Press this key to set communication settings.

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.

* On DL1540CL models with the internal hard disk (option), the half pitch interface connector is replaced with a SCSI interface connector. Thus, the RS-232-C unit cannot be connected.

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.

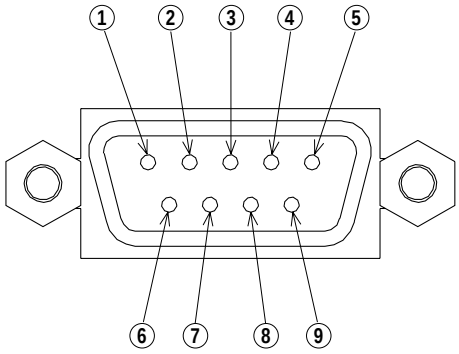
RS-232C Interface Specifications

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.

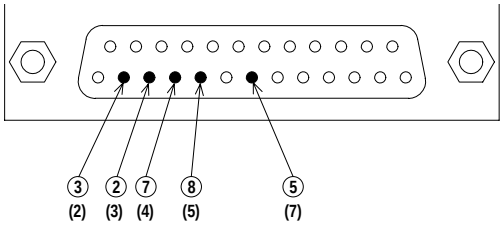
Connector and Signal Names



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.

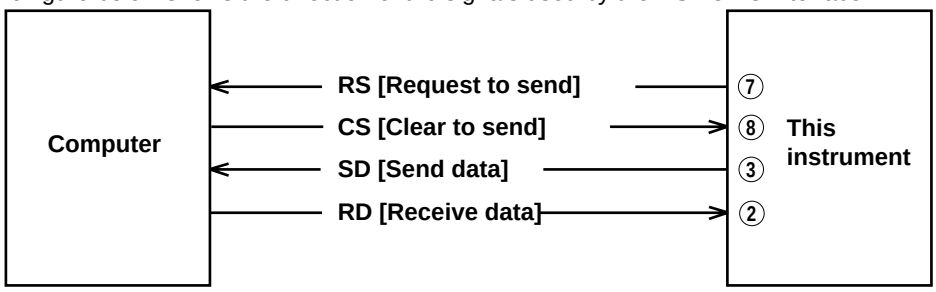


Table of RS-232-C Standard Signals and their JIS and CCITT Abbreviations

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		Software Handshake	Hardware Handshake	
	The menu of this instrument	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.	No handshake	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.	No handshake
OFF-OFF	NO-NO			○			○
XON-XON	XON-XON	○			○		
XON-RS	XON-RTS	○				○	
CS-RS	CTS-RTS		○			○	

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 intrument 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 intstruments 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 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.

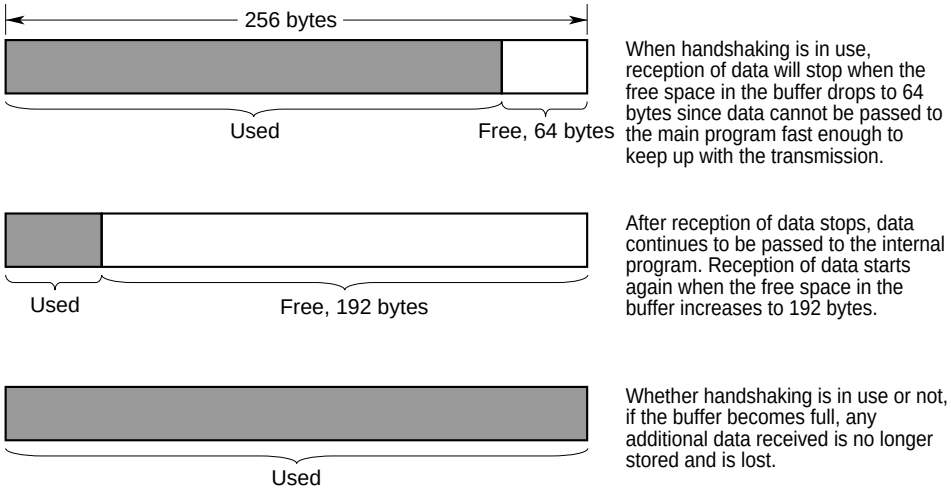
4 CS-RS

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

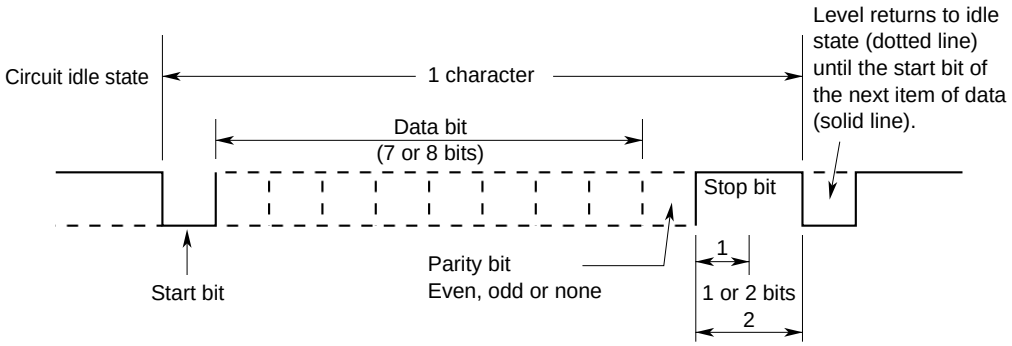
When handshaking is used to control the reception of data, data may still be sent from the computer even if the free space in the receive buffer drops below 64 bytes. In this case, after the receive buffer becomes full, the excess data will be lost, whether handshaking is in effect or not. Data storage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



Data Receiving Control using Handshaking

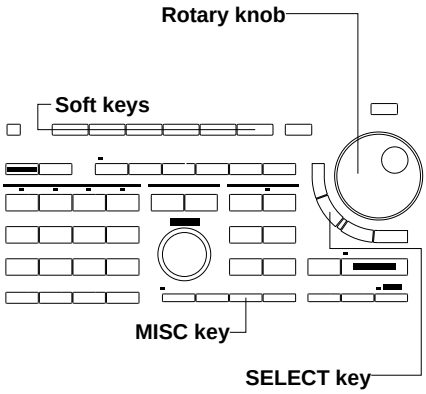
2.5 Matching the Data Format

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

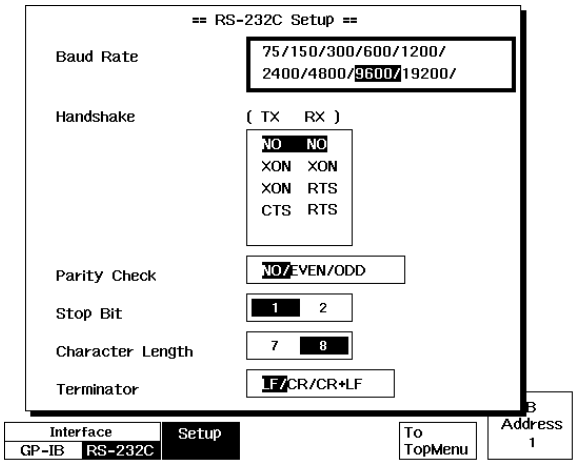
Keys and Procedure



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.

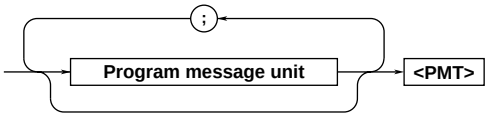
3 Before Programming

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.

Example : `TIMEBASE:SOURCE INTERNAL;TDIV 1ms<PMT>`
Unit Unit

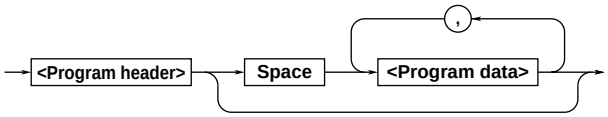
<PMT>

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.

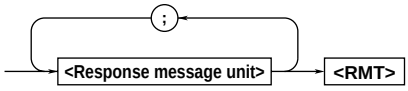
<Program data>

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.

Example : `TIMEBASE:SOURCE INTERNAL<PMT>`
Header Data

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>

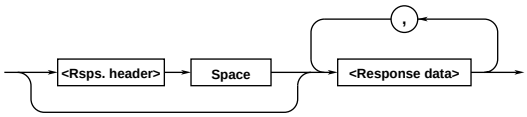
A response message consists of one or more response message units: each response message unit corresponds to one response.
Response message units are delimited by a ";".
For the response message format, refer to the next page.

Example : `FILTER:BANDWIDTH FULL;SMOOTHING 0<RMT>`
Unit Unit

<RMT>

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

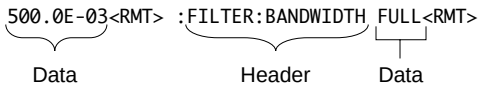
Response message unit format
The format of a program message unit is shown below.



<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 it 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.

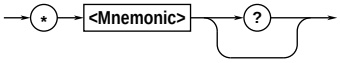
3.2 Commands

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

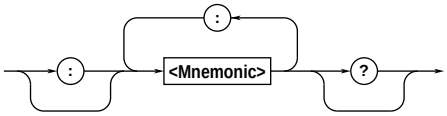
Commands defined in IEEE 488.2-1987 are called common commands. The header format of a common command is shown below. An asterisk (*) must always be attached to the beginning of a command.



An example of a common command
```
*CLS
```

Compound Header

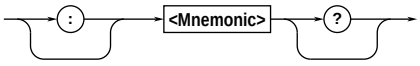
Commands designed to be used only with this instrument are classified and arranged in a hierarchy according to their function. The format of a compound header is illustrated below. A colon (:) must be used when specifying a lower-level header.



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 "`CHANnel<x>`" is written as "`CHAN`", this represents "`CHANnel1`".

- Any part of a command enclosed by [ ] can be omitted.
  Example `CHANnel<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 "`CHANnel1?`" and "`CHANnel1: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 →CHANnel2: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 →CHANnel2:INPut:COUPling {AC|DC|GND})
<Boolean>	Indicates ON/OFF. Set to ON, OFF or value (Example: CH2 display ON →CHANnel2: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 →#6000010ABCDEFGHIJ)

<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^{18}
PE	Peta	10^{15}
T	Tera	10^{12}
G	Giga	10^{9}
MA	Mega	10^{6}
K	Kilo	10^{3}
M	Mili	10^{-3}
U	Micro	10^{-6}
N	Nano	10^{-9}
P	Pico	10^{-12}
F	Femto	10^{-15}
A	Atto	10^{-18}

<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.

<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> <Data byte string>	#6000010ABCDEFGHIJ

#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 “0AH”, 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).

```
CHANnel1:VDIV:VALue 5V;VALue?<PMT>
```

This is because the "CHANnel1: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:EESE
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:EESE 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:EESE
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
:CURSor:FFT:P<x>?	Queries the P values of the FFT cursors	4-19
:CURSor:FFT:POSition<x>	Sets the position of the FFT cursor, or queries the current setting	4-19
:CURSor[:MODE]	Sets cursor measurement ON/OFF, or queries the current setting	4-19
:CURSor:TY?	Queries all the T-Y cursor measurement settings	4-19
:CURSor:TY:DDEGree?	Queries the phase difference value, ΔD, between D1 and D2 phase cursors	4-19
:CURSor:TY:DEGRee?	Queries the value of D1 or D2 cursor	4-19
:CURSor:TY:DT?	Queries the ΔT value between the T-Y cursors	4-19
:CURSor:TY:DV?	Queries the ΔV value between the T-Y cursors	4-20
:CURSor:TY:PERDt?	Queries the 1P/ΔT value between the T-Y cursors	4-20
:CURSor:TY:POSition<x>	Sets the position of the T-Y cursor, or queries the current setting	4-20
:CURSor:TY:REFerence<x>	Sets the position of the start point or end point (R1, R2) for the phase reference, or queries the current setting	4-20
:CURSor:TY:RVALue	Sets the phase reference value between the vertical cursors (phase cursors), or queries the current setting	4-20
:CURSor:TY:TRACe	Sets the waveform of measurement of the T-Y cursor, or queries the current setting	4-20
:CURSor:TY:TYPE	Sets the measurement type of the T-Y cursor, or queries the current setting	4-20
:CURSor:TY:T<x>?	Queries the time corresponding to the T-Y cursor	4-20
:CURSor:TY:V<x>?	Queries the voltage value corresponding to the T-Y cursor	4-20
:CURSor:XY?	Queries all the X-Y cursor measurement settings	4-20
:CURSor:XY:DX?	Queries the ΔX value between the X-Y cursors	4-20
:CURSor:XY:DY?	Queries the ΔY value between the X-Y cursors	4-20
:CURSor:XY:POSiton<x>	Sets the position of the X-Y cursor, or queries the current setting	4-20
:CURSor:XY:T?	Queries the time corresponding to the X-Y cursor	4-21
:CURSor:XY:TYPE	Sets the measurement type of the X-Y cursor, or queries the current setting	4-21
:CURSor:XY:X?	Queries the voltage value corresponding to the X waveform of the X-Y cursor	4-21
:CURSor:XY:Y?	Queries the voltage value corresponding to the Y waveform of the X-Y cursor	4-21
DISPlay Group		
:DISPlay?	Queries all the display settings	4-23
:DISPlay:ACCumulate?	Queries all the accumulation settings	4-23
:DISPlay:ACCumulate:COLor	Sets the color grade width of the waveform, or queries the current setting	4-23
:DISPlay:ACCumulate:MODE	Turns accumulation mode ON/OFF, or queries the current setting	4-23
:DISPlay:ACCumulate:PERSistance	Sets the accumulation time, or queries the current setting	4-23
:DISPlay:ALLexec	Displays all stored waveforms	4-23
:DISPlay:COUNt	Sets the number of the waveform to be displayed, or queries the current setting	4-23
:DISPlay:GRATicule	Sets the graticule, or queries the current setting	4-24
:DISPlay:INTENsity?	Queries the intensity of displayed parameters	4-24
:DISPlay:INTENsity:{CURSor|SCALe|SNAPshot|TEXT|WAVE}	Sets the intensity of each displayed parameter, or queries the current setting	4-24
:DISPlay:INTERpolate	Sets the interpolation method, or queries the current setting	4-24
:DISPlay:PMARker	Sets the % marker ON/OFF, or queries the current setting	4-24
:DISPlay:POSition	Sets the horizontal/vertical position of the waveform, or queries the current setting	4-24
:DISPlay:SCALe	Turns display of the scale ON/OFF, or queries the current setting	4-24
:DISPlay:TIME?	Queries the time of acquisition corresponding to the waveform with the display No.	4-24
:DISPlay:WINFormation	Turns display of waveform information ON/OFF, or queries the current setting	4-24
:DISPlay:WSCReen	Turns the wide screen mode ON/OFF, or queries the current setting	4-24
:DISPlay:XY	Turns display of the X-Y waveform ON/OFF, or queries the current setting	4-24
FILE Group		
:FILE?	Queries all the settings relating to files	4-26
:FILE:ANAMing	Sets the auto naming mode for file names, or queries the current setting	4-26
:FILE:COPY:ACQuisition:EXECute	Copies the acquisition data file on the internal hard disk to the floppy disk	4-27
:FILE:COPY:ACQuisition:FORM	Sets the data format of the waveform to be copied, or queries the current setting	4-27
:FILE:COPY:ACQuisition:NAME	Sets the name of the acquisition data file to be copied	4-27
:FILE:COPY:I2C:EXECute	Copies the data file containing the I^2C-bus analysis results on the internal hard disk to the floppy disk	4-27
:FILE:COPY:I2C:NAME	Sets the name of the file containing the I^2C-bus analysis results to be copied	4-27
:FILE:COPY:IMAGe:EXECute	Copies the screen image data file on the internal hard disk to the floppy disk	4-27
:FILE:COPY:IMAGe:FORMat	Sets the data format of the screen image data to be copied, or queries the current setting	4-27
:FILE:COPY:IMAGe:NAME	Sets the name of the screen image data file to be copied	4-27

Command	Description	Page
:FILE:COPY:SETup:EXECute	Copies the setting parameter file on the internal hard disk to the floppy disk	4-27
:FILE:COPY:SETup:NAME	Sets the name of the setting parameter file to be copied	4-27
:FILE:COPY:TRACe:EXECute	Copies the display waveform data file on the internal hard disk to the floppy disk	4-27
:FILE:COPY:TRACe:NAME	Sets the name of the display waveform data file to be copied	4-27
:FILE:DELete:ACQuisition:EXECute	Executes the deletion of an acquisition data file	4-27
:FILE:DELete:ACQuisition:FORM	Sets the data format of the waveform to be deleted, or queries the current setting	4-28
:FILE:DELete:ACQuisition:NAME	Sets the name of the acquisition data file to be deleted	4-28
:FILE:DELete:I2C:EXECute	Deletes the file containing the I^2C-bus analysis results	4-28
:FILE:DELete:I2C:NAME	Sets the name of the file containing the I^2C-bus analysis results to be deleted	4-28
:FILE:DELete:IMAGe:EXECute	Deletes the screen image data file	4-28
:FILE:DELete:IMAGe:FORMat	Sets the data format of the screen image data to be deleted, or queries the current setting	4-28
:FILE:DELete:IMAGe:NAME	Sets the name of the screen image data file to be deleted	4-28
:FILE:DELete:SETup:EXECute	Executes the deletion of a setting parameter data file	4-28
:FILE:DELete:SETup:NAME	Sets the name of the setting parameter file to be deleted	4-28
:FILE:DELete:TRACe:EXECute	Executes the deletion of a displayed waveform file	4-28
:FILE:DELete:TRACe:NAME	Sets the name of the displayed waveform file to be deleted	4-28
:FILE:DRIVe	Sets the target media to execute the FILE Group Commands, or queries the current setting	4-28
:FILE:FORMat?	Queries all the settings relating to floppy disk formatting	4-28
:FILE:FORMat:EXECute	Formats the floppy disk	4-28
:FILE:FORMat:TYPE	Selects the floppy disk format to be used for formatting, or queries the current setting	4-29
:FILE:INFormation	Checks floppy disk file information	4-29
:FILE:LOAD?	Queries the settings relating to loading data	4-29
:FILE:LOAD:ACQuisition:EXECute	Executes the loading of acquisition waveform data	4-29
:FILE:LOAD:ACQuisition:NAME	Sets the name of the acquisition waveform data file to be loaded	4-29
:FILE:LOAD:SETup:EXECute	Loads setting parameters	4-29
:FILE:LOAD:SETup:NAME	Sets the name of the setting parameter file to be loaded	4-29
:FILE:LOAD:TRACe?	Queries all the setting values relating to loading P-P waveform data	4-29
:FILE:LOAD:TRACe:EXECute	Loads P-P waveform data	4-29
:FILE:LOAD:TRACe:NAME	Sets the name of the P-P waveform data file to be loaded	4-29
:FILE:LOAD:TRACe:TRACe	Sets the trace No. where the P-P waveform data will be loaded	4-29
:FILE:PROTect:ACQuisition:EXECute	Protects acquisition data	4-29
:FILE:PROTect:ACQuisition:FORM	Sets the data format of the waveform to be protected, or queries the current setting	4-29
:FILE:PROTect:ACQuisition:NAME	Sets the name of the acquisition waveform data file to be protected	4-29
:FILE:PROTect:I2C:EXECute	Protects the file containing the I^2C-bus analysis results	4-29
:FILE:PROTect:I2C:NAME	Sets the name of the file containing the I^2C-bus analysis results to be protected	4-29
:FILE:PROTect:IMAGe:EXECute	Protects the screen image data file	4-30
:FILE:PROTect:IMAGe:FORMat	Sets the data format of the screen image data to be protected, or queries the current setting	4-30
:FILE:PROTect:IMAGe:NAME	Sets the name of the screen image data file to be protected	4-30
:FILE:PROTect:SETup:EXECute	Protects setting parameters	4-30
:FILE:PROTect:SETup:NAME	Sets the name of the setting parameter file to be protected	4-30
:FILE:PROTect:TRACe:EXECute	Protects P-P waveform data	4-30
:FILE:PROTect:TRACe:NAME	Sets the name of the P-P waveform data file to be protected	4-30
:FILE:SAVE?	Queries all settings relating to saving data	4-30
:FILE:SAVE:ACQuisition:EXECute	Saves acquisition waveform data	4-30
:FILE:SAVE:ACQuisition:FORM	Sets data format for waveform data save, or queries current setting	4-30
:FILE:SAVE:ACQuisition:NAME	Sets the name of the acquisition waveform data file to be saved	4-30
:FILE:SAVE:SETup:EXECute	Saves setting parameters	4-30
:FILE:SAVE:SETup:NAME	Sets the name of the setting parameter file to be saved	4-30
:FILE:SAVE:TRACe?	Queries all settings relating to saving P-P waveform data	4-30
:FILE:SAVE:TRACe:EXECute	Saves P-P waveform data	4-30
:FILE:SAVE:TRACe:NAME	Sets the name of the P-P waveform data file to be saved	4-30
:FILE:SAVE:TRACe:TRACe	Selects the trace No. of the P-P waveform to be saved, or queries the current setting	4-30
FILTer Group		
:FILTer?	Queries all the settings relating to the input filter	4-31
:FILTer:BANDwidth	Sets the bandwidth limits, or queries the current setting	4-31
:FILTer:SMOothing	Turns smoothing ON/OFF, or queries the current setting	4-31

Command	Description	Page
GONogo Group		
:GONogo?	Queries all the GO/NO-GO settings	4-34
:GONogo[:MODE]	Turns GO/NO-GO determination ON/OFF, or queries the current setting	4-34
:GONogo:PARameter?	Queries all the settings relating to GO/NO-GO determination which use parameters	4-34
:GONogo:PARameter:ABORt	Aborts parameter GO/NO-GO determination	4-34
:GONogo:PARameter:ACTion	Sets the action to be taken if the parameter determination result is NO-GO, or queries the current setting	4-34
:GONogo:PARameter:BUZZer	Sets the ON/OFF setting of the buzzer when NO-GO occurs during patameter GO/NO-GO determination, or queries the current setting	4-34
:GONogo:PARameter:COUNt?	Queries the number of times parameter GO/NO-GO is performed	4-34
:GONogo:PARameter:EXECute	Executes parameter GO/NO-GO determination	4-34
:GONogo:PARameter:FNAMe	Selects the file name of the NG data to be output	4-34
:GONogo:PARameter:ITEM<x>?	Queries all the settings relating to the specified waveform parameter for parameter GO/NO-GO determination	4-35
:GONogo:PARameter:ITEM<x>:CAUSe?	Queries whether the specified waveform parameter used for parameter GO/NO-GO determination is the cause of failure	4-35
: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	4-35
: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	4-35
: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	4-35
:GONogo:PARameter:ITEM<x>:TYPE:<parameter>	Sets the upper/lower limits for the specified measurement item, or queries the current setting	4-35
:GONogo:PARameter:ITEM<x>:VALue?	Queries the measured value of the specified parameter used for parameter GO/NO-GO determination	4-35
:GONogo:PARameter:LOGic	Sets the determination logic for parameter GO/NO-GO, or queries the current setting	4-35
:GONogo:PARameter:NGCount?	Queries the number of NGs occurring during parameter GO/NO-GO determination	4-35
:GONogo:PARameter:SEQuence	Sets the number of times parameter GO/NO-GO determination is performed, or queries the current setting	4-36
:GONogo:PARameter:STOPcount	Sets the number of times to execute parameter determination, or queries the current setting	4-36
:GONogo:PARameter:TRANge	Sets the parameter GO/NO-GO determination range, or queries the current setting	4-36
:GONogo:PARameter:WAIT?	Wait for parameter determination to finish	4-36
:GONogo:TYPE	Sets the GO/NO-GO type, or queries the current setting	4-36
:GONogo:ZONE?	Queries all the settings relating to zone GO/NO-GO determination	4-36
:GONogo:ZONE:ABORt	Aborts GO/NO-GO determination	4-36
:GONogo:ZONE:ACTion	Sets the action to be taken if the zone determination is NG, or queries the current setting	4-36
: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	4-36
:GONogo:ZONE:COUNt?	Queries the number of times zone GO/NO-GO determination is performed	4-36
:GONogo:ZONE:EDIT<x>:EXIT	Exits from the edit menu for the specified zone	4-36
:GONogo:ZONE:EDIT<x>:PART	Selects partial edit for the specified zone during zone GO/NO-GO determination	4-37
:GONogo:ZONE:EDIT<x>:WHOLe	Selects total edit for the specified zone	4-37
:GONogo:ZONE:EXECute	Executes zone determination	4-37
:GONogo:ZONE:FNAMe	Selects the file name of the NO-GO data to be output	4-37
:GONogo:ZONE:LOGic	Sets the determination logic for zone GO/NO-GO determination, or queries the current settings	4-37
:GONogo:ZONE:NGCount?	Queries the number of NGs occurring during zone GO/NO-GO determination	4-37
:GONogo:ZONE:SEQuence	Sets the number of times GO/NO-GO determination is performed, or queries the current setting	4-37
:GONogo:ZONE:STOPcount	Sets the number of times to execute zone determination, or queries the current setting	4-37
:GONogo:ZONE:TRACe<x>?	Queries all the settings made for the specified zone for zone GO/NO-GO determination	4-37
:GONogo:ZONE:TRACe<x>:CAUSe?	Queries whether the specified trace used for zone GO/NO-GO determination is the cause of failure	4-37

Command	Description	Page
:GONogo:ZONE:TRACe<x>:CONDition	Sets the criteria for the specified trace used for zone GO/NO-GO determination, or queries the current setting	4-38
:GONogo:ZONE:TRACe<x>:MODE	Selects whether the zone determination result for the specified trace is validated or not, or queries the current setting	4-38
:GONogo:ZONE:TRACe<x>:ZONE	Sets the zone data for each trace, or queries the current setting	4-38
:GONogo:ZONE:WAIT?	Wait for zone determination to finish	4-38
HCOPy Group		
:HCOPy?	Queries all the settings relating to output of screen data	4-42
:HCOPy:COMMent?	Queries all the settings relating to the comment to be printed	4-42
:HCOPy:COMMent:CONTents	Sets the comment to be printed	4-42
:HCOPy:COMMent:MODE	Turns the printing of comments ON/OFF, or queries the current setting	4-42
:HCOPy:EQUipment	Selects the device to which the screen data is to be output, or queries the current setting	4-43
:HCOPy:EXTPrinter?	Queries all the settings relating to output an external printer	4-43
:HCOPy:EXTPrinter:FORMat	Sets the data format for printout to an external printer, or queries the current setting	4-43
:HCOPy:EXTPrinter:RESolution	Sets the output resolution when printing in BJ format, or queries the current setting	4-43
:HCOPy:EXTPrinter:TONE	Sets the half tone of the external printer output, or queries the current setting	4-43
:HCOPy:FLOPpy?	Queries all the settings relating to saving of screen image data on a floppy disk	4-43
: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	4-43
:HCOPy:FLOPpy:COMPression	Set the compression of the screen image data to output ON/OFF, or queries the current setting	4-43
:HCOPy:FLOPpy:EXECute	Saves the screen image data	4-43
:HCOPy:FLOPpy:FORMat	Selects the format of the screen image data to be saved, or queries the current setting	4-43
:HCOPy:FLOPpy:NAME	Selects the name of the screen image data file to be output	4-43
:HCOPy:FLOPpy:PEN?	Queries all the settings relating to the pens of the image data to be output	4-44
:HCOPy:FLOPpy: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	4-44
:HCOPy:FLOPpy:PEN:MANual?	Queries the number of pens of the image data to be output in case pens are assigned automatically	4-44
:HCOPy:FLOPpy:PEN:MANual:GRID	Sets the pen Nos. for the graticule etc. in case pens are assigned manually, or queries the current setting	4-44
:HCOPy:FLOPpy:PEN:MANual:LOAD<x>	Sets the pen Nos. of all loaded traces in case pens are assigned manually, or queries the current setting	4-44
:HCOPy:FLOPpy:PEN:MANual:TRACe<x>	Sets the pen No. for the specified trace in case pens are assigned manually, or queries the current setting	4-44
:HCOPy:FLOPpy:PEN:MODE	Sets the pen assignment method, or queries the current setting	4-44
:HCOPy:FLOPpy:PFORmat	Sets the output size for the screen image data, or queries the current setting	4-44
:HCOPy:FLOPpy:SPEed	Sets the pen speed of the screen image data to be output, or queries the current setting	4-44
:HCOPy:FLOPpy:TONE	Set the color tone of the screen image data to output, or queries the current setting	4-44
:HCOPy:FLOPpy:XYMode	Selects the plot method for X-Y waveforms, or queries the current setting	4-44
:HCOPy:HD?	Queries the setting parameters related to saving of the screen image data to the internal hard disk	4-45
:HCOPy:HD:ANAMing	Sets the auto naming mode for the file name of the screen image data to be output to the internal hard disk, or queries the current setting	4-45
:HCOPy:HD:COMPression	Turns ON/OFF the compression for the screen image data output, or queries the current setting	4-45
:HCOPy:HD:EXECute	Saves the screen image data to the internal hard disk	4-45
:HCOPy:HD:FORMat	Selects the format of the screen image data to be saved to the internal hard disk, or queries the current setting	4-45
:HCOPy:HD:NAME	Selects the name of the screen image data file to be output to the internal hard disk	4-45
:HCOPy:HD:PEN?	Queries all the settings related to the pens used to output the image data to the internal hard disk	4-45
:HCOPy:HD:PEN:AUTO	Sets the number of pens when the pens used to output the screen image data to the internal hard disk are assigned automatically, or queries the current setting	4-45
:HCOPy:HD:PEN:MANual?	Queries all settings when the pens used to output the screen image data to the internal hard disk are assigned manually	4-45

Command	Description	Page
:HCOPy:SCSI:PEN:MANual:TRACe<x>	Sets the pen No. for the specified trace when pens are assigned manually, or queries the current setting	4-49
:HCOPy:SCSI:PEN:MODE	Sets the pen assignment method, or queries the current setting	4-49
:HCOPy:SCSI:PFORmat	Sets the output size for the screen image data, or queries the current setting	4-50
:HCOPy:SCSI:SPEed	Sets the pen speed, or queries the current setting	4-50
:HCOPy:SCSI:TONE	Set the color tone of the screen image data to output, or queries the current setting	4-50
:HCOPy:SCSI:XYMode	Selects the plot method for X-Y waveforms, or queries the current setting	4-50
IMAGe Group		
:IMAGe?	Queries all screen-image data output settings	4-51
:IMAGe:FORMat	Sets/queries screen-image data output format	4-51
:IMAGe:SEND?	Queries screen-image data	4-51
:IMAGe:TONE	Set the color tone of the screen image data to output, or queries the current setting	4-51
INITialize Group		
:INITialize:EXECute	Executes initialization	4-51
MATH Group		
:MATH?	Queries all the settings relating to waveform computation	4-52
:MATH:FFT?	Queries all the settings relating to FFT computation	4-52
:MATH:FFT:DISPlay	Sets the FFT waveform to be displayed, or queries the current setting	4-52
:MATH:FFT:MODE	Sets the FFT waveform display ON/OFF, or queries the current setting	4-52
:MATH:FFT:TRACe	Sets the trace for FFT measurement, or queries the current setting	4-53
:MATH:FFT:WINDow	Sets the FFT window function, or queries the current setting	4-53
:MATH:OPERator	Sets the computation contents	4-53
:MATH:PHASe?	Queries all the settings relating to phase shifts	4-53
:MATH:PHASe[:SHIFt]	Sets the desired phase shift, or queries the current setting	4-53
:MATH:SENSitivity	Sets the display sensitivity of the vertical axis, or queries the current setting	4-53
MEASure Group		
:MEASure?	Queries all the settings relating to automatic measurement of waveform parameters	4-56
:MEASure:ALLScan:ABORt	Aborts automatic measurement of acquisition data	4-56
:MEASure:ALLScan:EXECute	Starts automatic measurement of acquisition data	4-56
:MEASure:{LOAD<x>|TRACe<x>}?	Queries all the settings relating to the measurement items for each waveform	4-57
:MEASure:{LOAD<x>|TRACe<x>}:ALL	Sets the parameters for the specified waveform ON/OFF	4-57
:MEASure:{LOAD<x>|TRACe<x>}:COPY	Copies the ON/OFF settings of all measurement items of a specific channel to all other waveforms	4-57
:MEASure:{LOAD<x>|TRACe<x>}:DELay?	Queries all the settings relating to delay measurement between channels for a specified waveform	4-57
: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	4-57
: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	4-57
:MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:LEVel?	Queries all the settings relating to the threshold level	4-57
:MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:LEVel:PERCent	Sets the % value of the threshold level, or queries the current setting	4-57
:MEASure:{LOAD<x>|TRACe<x>}:DELay:MEASure:LEVel:UNIT	Sets the voltage value of the threshold level, or queries the current setting	4-58
: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	4-58
:MEASure:{LOAD<x>|TRACe<x>}:DELay[:MODE]	Sets the delay measurement between channels for a specified waveform ON/OFF, or queries the current setting	4-58
:MEASure:{LOAD<x>|TRACe<x>}:DELay:REFerence?	Queries all settings relating to the reference waveform of delay measurement between channels for a specified waveform	4-58

Command	Description	Page
: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	4-58
: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	4-58
: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	4-58
: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	4-59
:MEASure:{LOAD<x>|TRACe<x>}:DELay:REFerence:SLOPe		
	Sets the slope of the reference waveform, or queries the current setting	4-59
:MEASure:{LOAD<x>|TRACe<x>}:DELay:REFerence:TRACe		
	Sets the trace of the reference waveform, or queries the current setting	4-59
:MEASure:{LOAD<x>|TRACe<x>}:DPRoximal?		
	Queries all settings relating to distal, mesial, and proximal points	4-59
:MEASure:{LOAD<x>|TRACe<x>}:DPRoximal:PERCent		
	Sets the distal, mesial, and proximal points in percentages, or queries the current settings	4-59
:MEASure:{LOAD<x>|TRACe<x>}:DPRoximal:UNIT		
	Sets the distal, mesial, and proximal points in terms of voltages, or queries the current settings	4-59
:MEASure[:MODE]	Turns automated measurements ON/OFF, or queries the current setting	4-59
:MEASure:{LOAD<x>|TRACe<x>}:<*parameter*>		
	Turns a measurement item ON/OFF for a specified waveform, or queries the current setting	4-59
:MEASure:TRANge	Sets the measurement range, or queries the current setting	4-60
:MEASure:{LOAD<x>|TRACe<x>}:VALue? <*parameter*>		
	Queries the automatically measured value of the specified measurement item for the specified waveform	4-60
:MEASure:WAIT?	Wait for automatic measurement to finish	4-60
RECall Group		
:RECall?	Queries all settings relating to the recalling of data from the internal memory	4-61
:RECall:LOAD<x>	Sets the display of the loaded waveform ON/OFF, or queries the current setting	4-61
:RECall:MEMory<x>	Recalls waveform data from the specified internal memory and recalls it as a specified trace	4-61
:RECall:POSition<x>	Sets the vertical position of the recalled waveform, or queries the current setting	4-61
:RECall:SETup	Recalls the specified set-up data from the internal memory	4-61
SCAle Group		
:SCALe?	Queries all the settings relating to linear scaling	4-62
:SCALe:CHANnel<x>?	Queries all the settings relating to linear scaling for a specified channel	4-62
:SCALe:CHANnel<x>:A	Sets the linear scaling A value for a specified channel, or queries the current setting	4-63
:SCALe:CHANnel<x>:B	Sets the linear scaling B value for a specified channel, or queries the current setting	4-63
:SCALe:CHANnel<x>:MODE	Sets linear scaling ON/OFF, or queries the current setting	4-63
:SCALe:CHANnel<x>:UNIT	Sets the linear scaling unit for a specific channel, or queries the current setting	4-63
:SCALe:LOAD<x>?	Queries all the settings relating to linear scaling for a specified loaded waveform	4-63
:SCALe:LOAD<x>:A	Sets the linear scaling A value for a specified loaded waveform, or queries the current setting	4-63
:SCALe:LOAD<x>:B	Sets the linear scaling B value for a specified loaded waveform, or queries the current setting	4-63
:SCALe:LOAD<x>:MODE	Sets linear scaling for a specified loaded waveform ON/OFF, or queries the current setting	4-63
:SCALe:LOAD<x>:UNIT	Sets the linear scaling unit for a specified loaded waveform, or queries the current setting	4-63
:SCALe:MODE	Sets linear scaling ON/OFF, or queries the current setting	4-63

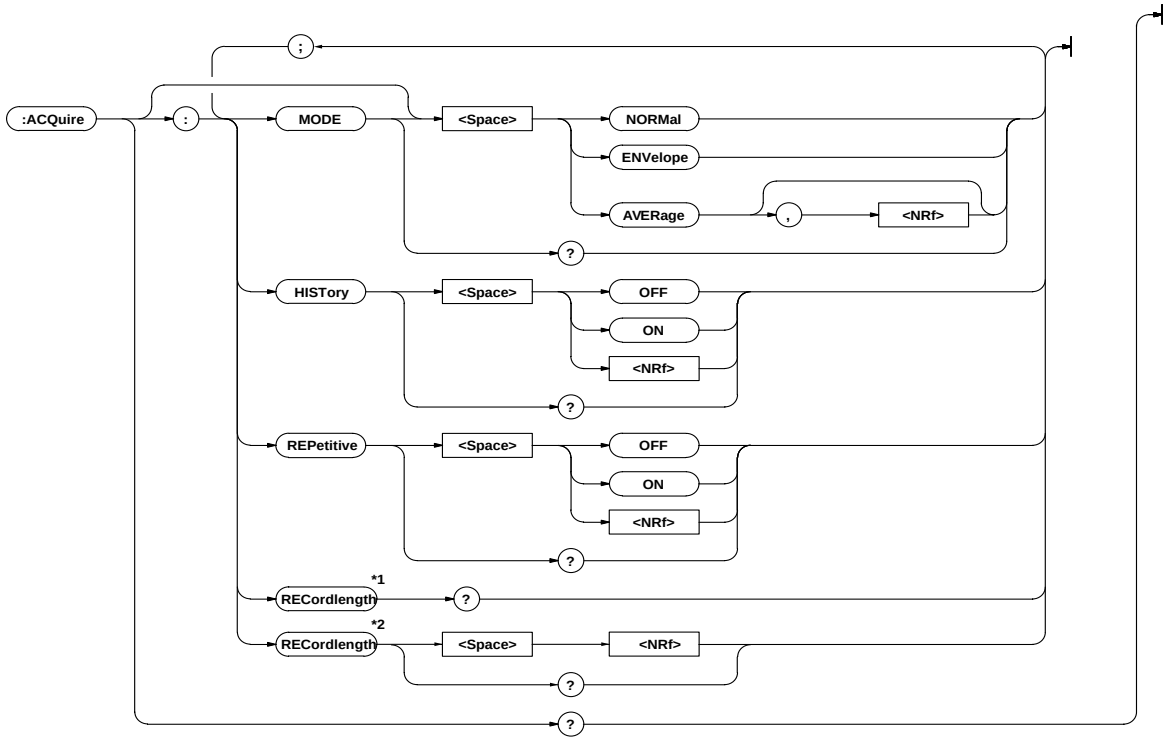
Command	Description	Page
SNAP Group		
:SNAP	Executes a snapshot	4-64
STARt Group		
:STARt	Starts waveform acquisition	4-64
STATus Group		
:STATus?	Queries all the settings relating to the status of the communication function	4-65
:STATus:CONDition?	Queries the contents of the condition register	4-65
:STATus:EESE	Sets the extended event enable register, or queries the current setting	4-65
:STATus:EESR?	Queries the contents of the extended event register and clears the register	4-65
:STATus:ERRor?	Queries the code and the message of the error which has occurred	4-65
:STATus:FILTer<x>	Queries all the settings relating to the specified transition filter, or queries the current setting	4-65
:STATus:QENable	Selects whether or not to store a message other than an error message into the error queue, or queries the current setting	4-65
:STATus:QMESsage	Selects whether or not to add the message contents to a response to “STATus:ERRor?”, or queries the current setting	4-65
:STATus:SPOLL?	Executes a serial poll	4-65
STOP Group		
:STOP	Stops waveform acquisition	4-65
STORe Group		
:STORe:SETup	Stores setting parameters in the internal memory	4-66
:STORe:TRACe<x>	Stores waveform data in the internal memory	4-66
SYSTem Group		
:SYSTem?	Queries all the system settings	4-67
:SYSTem:BATTery?	Queries the condition of the internal lithium battery	4-67
:SYSTem:CLOCk?	Queries all the settings relating to the date and time	4-67
:SYSTem:CLOCk:DATE	Sets the date, or queries the current setting	4-67
:SYSTem:CLOCk:MODE	Turns display of the date and time ON/OFF, or queries the current setting	4-67
:SYSTem:CLOCk:TIME	Sets the time, or queries the current setting	4-67
:SYSTem:CONFig?	Queries all the settings relating to the configuration system	4-67
:SYSTem:CONFig:EXTernalid	Sets the SCSI-ID number of the external SCSI device, or queries the current setting	4-67
:SYSTem:CONFig:HDMotor	Turns ON/OFF the internal hard disk (option) motor, or queries the current setting	4-67
:SYSTem:CONFig:PDENsity	Selects the level of the printing density, or queries the current setting	4-67
:SYSTem:LCD?	Queries all settings relating to the LCD	4-67
:SYSTem:LCD:BRIGhtness	Sets the brightness of the LCD, or queries the setting	4-67
:SYSTem:LCD:MODE	Turns the back light of the LCD ON/OFF, or queries the current setting	4-67
TIMebase Group		
:TIMebase?	Queries all the timebase settings	4-68
:TIMebase:SOURce	Sets the timebase, or queries the current setting	4-68
:TIMebase:SRATe?	Queries the current sample rate	4-68
:TIMebase:TDIV	Set the time axis, or queries the current setting	4-68
TRIGger Group		
:TRIGger?	Queries all trigger settings	4-71
:TRIGger:ACTion?	Queries all the settings relating to action-on-trigger	4-71
:TRIGger:ACTion:FNAMe	Sets the file name when saving upon an action-on-trigger	4-72
:TRIGger:ACTion:MODE	Selects the action-on-trigger mode, or queries the current setting	4-72
:TRIGger:ACTion:STARt	Starts the action-on-trigger	4-72
:TRIGger:ACTion:STOP	Stops the action-on-trigger	4-72
:TRIGger:MODE	Sets the trigger mode, or queries the current setting	4-72
:TRIGger:POSition	Sets the trigger position, or queries the current setting	4-72
:TRIGger:SIMPle?	Queries all trigger settings	4-72
:TRIGger[:SIMPle]:DELay	Sets the delay time for trigger, or queries the current setting	4-72
:TRIGger[:SIMPle]:EDGE?	Queries all the settings when an edge trigger is selected	4-72
:TRIGger[:SIMPle]:EDGE:SLOPe	Sets the trigger slope in case of an edge trigger, or queries the current setting	4-72
:TRIGger[:SIMPle]:EDGE:SOURce	Sets the trigger source in case of an edge trigger, or queries the current setting	4-73
:TRIGger[:SIMPle]:FUNCtion	Sets the type of trigger, or queries the current setting	4-73

Command	Description	Page
:TRIGger[:SIMPle]:HOLDoff?	Queries all the holdoff settings for a trigger	4-73
:TRIGger[:SIMPle]:HOLDoff:MODE	Turns holdoff ON/OFF for a trigger, or queries the current setting	4-73
:TRIGger[:SIMPle]:HOLDoff:TIME	Sets the holdoff time for a trigger, or queries the current setting	4-73
:TRIGger[:SIMPle]:OR?	Queries all the settings when an OR trigger is selected	4-73
:TRIGger[:SIMPle]:OR:CHANnel<x>	Setst the slope and state when an OR trigger is selected, or queries the current setting	4-73
:TRIGger[:SIMPle]:PATTern?	Queries all the settings when a pattern trigger is selected	4-73
:TRIGger[:SIMPle]:PATTern:CHANnel<x>	Sets the state for the specified channel for a pattern trigger, or queries the current setting	4-73
:TRIGger[:SIMPle]:PATTern:CLOCk?	Queries all the clock settings when a pattern trigger is selected	4-73
:TRIGger[:SIMPle]:PATTern:CLOCk:SLOPe	Sets the slope for the clock channel for a pattern trigger, or queries the current setting	4-73
:TRIGger[:SIMPle]:PATTern:CLOCk:SOURce	Sets the clock channel for a pattern trigger, or queries the current setting	4-74
:TRIGger[:SIMPle]:PATTern:CONDition	Sets the trigger condition for a pattern trigger, or queries the current setting	4-74
:TRIGger[:SIMPle]:TV?	Queries all the settings when a TV trigger has been selected	4-74
:TRIGger[:SIMPle]:TV:FIELd	Sets the field No. for a TV trigger, or queries the current setting	4-74
:TRIGger[:SIMPle]:TV:FRAMe	Sets the frame skip function when the trigger is set to TV trigger, or queries the current setting	4-74
:TRIGger[:SIMPle]:TV:LEVel	Sets the trigger level, or queries the current setting	4-74
:TRIGger[:SIMPle]:TV:LINE	Sets the line No., or queries the current setting	4-74
:TRIGger[:SIMPle]:TV:POLarity	Sets the polarity, or queries the current setting	4-74
:TRIGger[:SIMPle]:TV:TYPE	Sets the broadcasting system, or queries the current setting	4-74
:TRIGger[:SIMPle]:WIDTh?	Queries all the settings when a pulse trigger is selected	4-74
:TRIGger[:SIMPle]:WIDTh:CONDition	Sets the trigger condition for a pulse trigger, or queries the current setting	4-74
:TRIGger[:SIMPle]:WIDTh:SLOPe	Sets the trigger state for a pulse trigger, or queries the current setting	4-75
:TRIGger[:SIMPle]:WIDTh:SOURce	Sets the trigger source for a pulse trigger, or queries the current setting	4-75
:TRIGger[:SIMPle]:WIDTh:TIME	Sets the pulse width for a pulse trigger, or queries the current setting	4-75
:TRIGger[:SIMPle]:WINDow?	Queries all settings relating to the window trigger	4-75
:TRIGger[:SIMPle]:WINDow:CONDition	Sets the trigger condition for a window trigger, or queries the current setting	4-75
:TRIGger[:SIMPle]:WINDow:LEVel	Sets the center level for a window trigger, or queries the current setting	4-75
:TRIGger[:SIMPle]:WINDow:WIDTh	Sets the width of the window, or queries the current setting	4-75
:TRIGger:SOURce?	Queries all the settings relating to the trigger source	4-75
:TRIGger:SOURce:CHANnel<x>?	Queries all the trigger source settings for the specified channel	4-75
:TRIGger:SOURce:CHANnel<x>:LEVel	Sets the trigger level, or queries the current setting	4-75
:TRIGger:SOURce:COUPling	Sets the trigger coupling, or queries the current setting	4-76
:TRIGger:SOURce:EXTernal?	Queries the external trigger settings	4-76
:TRIGger:SOURce:EXTernal:LEVel	Sets the external trigger level, or queries the current setting	4-76
:TRIGger:SOURce:HFRejection	Turns the HF rejection ON/OFF, or queries the current setting	4-76
UNIT Group		
:UNIT?	Queries the default values of all units	4-77
:UNIT:FREQuency	Sets the default value for <Frequency>, or queries the current setting	4-77
:UNIT:TIME	Sets the default value for <Time>, or queries the current setting	4-77
:UNIT:VOLTage	Sets the default value for <Voltage>, or queries the current setting	4-77
WAVeform Group		
:WAVeform?	Queries all information relating to the waveform data	4-78
:WAVeform:BITS?	Queries the bit length of the waveform data	4-78
:WAVeform:BYTeorder	Sets the transmission order when words consist of more than 2 bytes, or queries the current setting	4-78
:WAVeform:DATaselect	Selects whether acquisition data is ACQ waveform data or P-P waveform data, or queries the current setting	4-78
:WAVeform:END	Sets the point at which the last item is to be located, or queries the current setting	4-79
:WAVeform:FORMat	Selects the format in which the data is to be formatted, or queries the current setting	4-79

Command	Description	Page
:WAVeform:LENGth?	Queries the number of records in the waveform	4-79
:WAVeform:OFFSet?	Queries the offset voltage of the waveform	4-79
:WAVeform:POSition?	Queries the vertical-axis position	4-79
:WAVeform:SEND?	Queries the waveform data	4-79
:WAVeform:SIGN?	Queries whether the data is signed or unsigned when binary data is used	4-79
:WAVeform:SRATe?	Queries the sample rate of the waveform	4-79
:WAVeform:STARt	Sets the point at which the first item is to be located, or queries the current setting	4-80
:WAVeform:TRACe	Sets the target waveform, or queries the current setting	4-80
:WAVeform:TRIGger?	Queries the points elapsed from the start point until the trigger point	4-80
:WAVeform:TYPE?	Queries the type of ACQ waveform data	4-80
:WAVeform:VDIV?	Queries the V/div value	4-80
ZOOM Group		
:ZOOM?	Queries all the settings relating to waveform expansion	4-80
:ZOOM:HPOSition	Sets the center position for time axis expansion, or queries the current setting	4-80
:ZOOM:HZOom	Sets the expansion rate in the horizontal direction, or queries the current setting	4-80
:ZOOM[:MODE]	Sets the display type for the expanded waveform, or queries the current setting	4-80
Common Command Group		
*CAL?	Performs calibration and queries the result	4-81
*CLS	Clears the standard event register, extended event register and error queue	4-81
*ESE	Sets the value for the standard event enable register, or queries the current setting	4-81
*ESR?	Queries the value of the standard event register and clears it at the same time	4-82
*IDN?	Queries the instrument model	4-82
*OPC	Sets the OPC event after execution of the overlap command	4-82
*OPC?	Queries whether the overlap command has been executed	4-82
*OPT?	Queries installed options	4-82
*PSC	Selects whether or not to clear registers when power is turned ON, or queries the current setting	4-82
*RST	Resets the current settings	4-82
*SRE	Sets the value of the service request enable register, or queries the current setting	4-82
*STB?	Queries the value of the status byte register	4-83
*TST?	Executes a self-test and queries the result	4-83
*WAI	Waits for the command following "*WAI" until execution of the designated overlap command has been completed	4-83

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^n)`
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

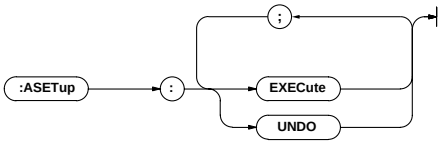
Function Sets the record length, or queries the current setting.
Syntax `ACQuire:RECordlength {<NRf>}`
`ACQuire:RECordlength?`
`<NRf>=1000,10000,100000,400000,1000000,2000000`
Example `ACQUIRE:RECORDLENGTH 10000`
`ACQUIRE:RECORDLENGTH?→:ACQUIRE:RECORDLENGTH 10000`
Description Available on DL1540CL.

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 settins 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

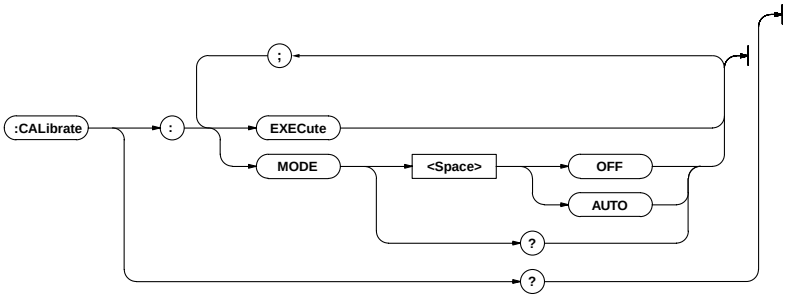
Function Executes auto set-up.
Syntax `ASETup:EXECute`
Example `ASETUP:EXECUTE`

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

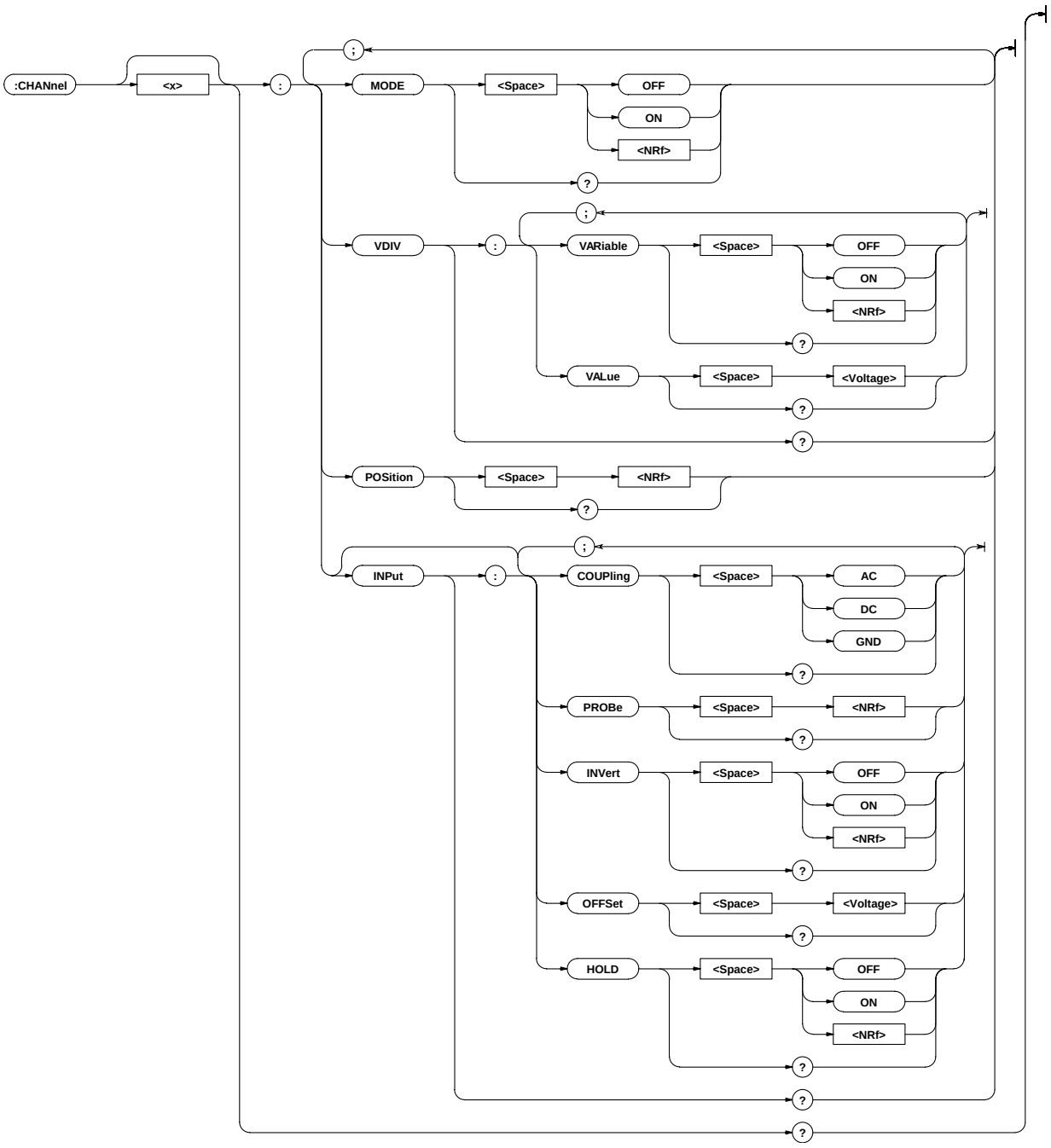
Function Executes calibration.
Syntax `CALibrate:EXECute`
Example `CALIBRATE:EXECUTE`

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>:POSition

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

Syntax
```
CHANnel<x>:POSition {<NRf>}
CHANnel<x>:POSition?
<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

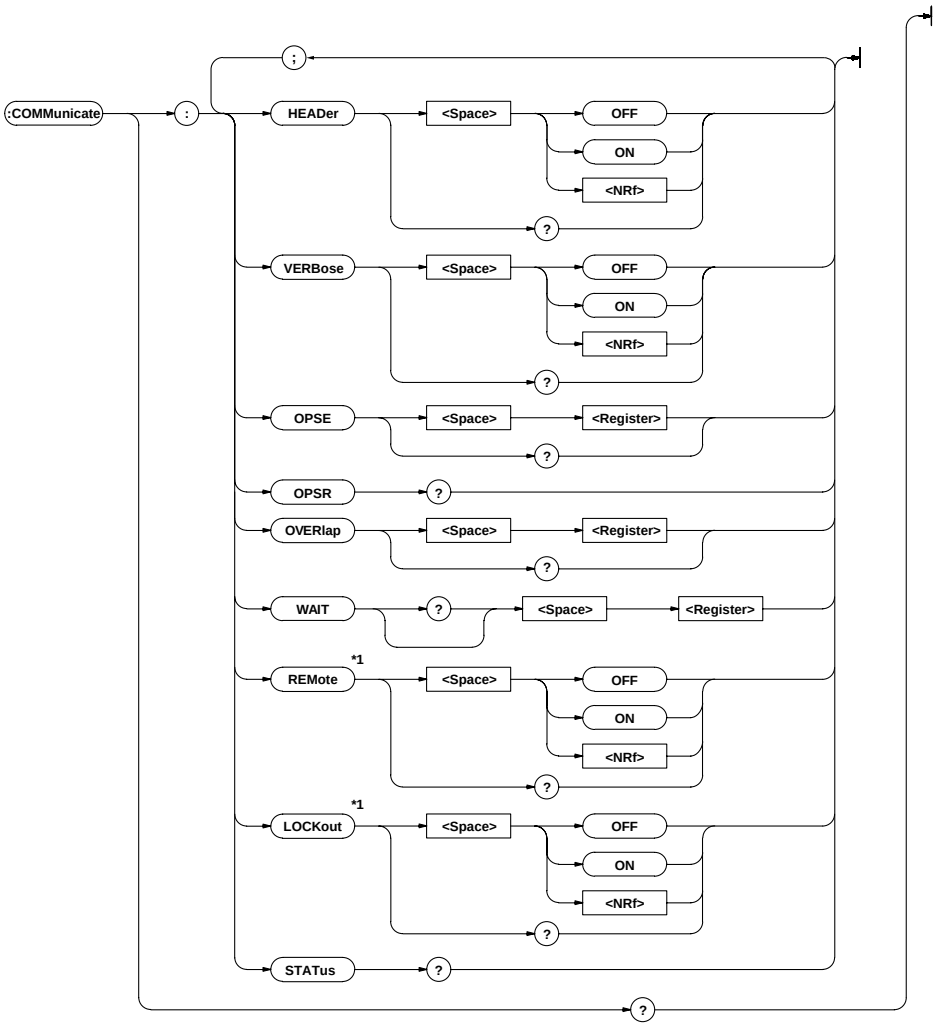
The CLEar command is used to clear the trace. This allows you to perform the same operation which you can perform using the CLEAR TRACE key on the front panel.

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. Ther 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

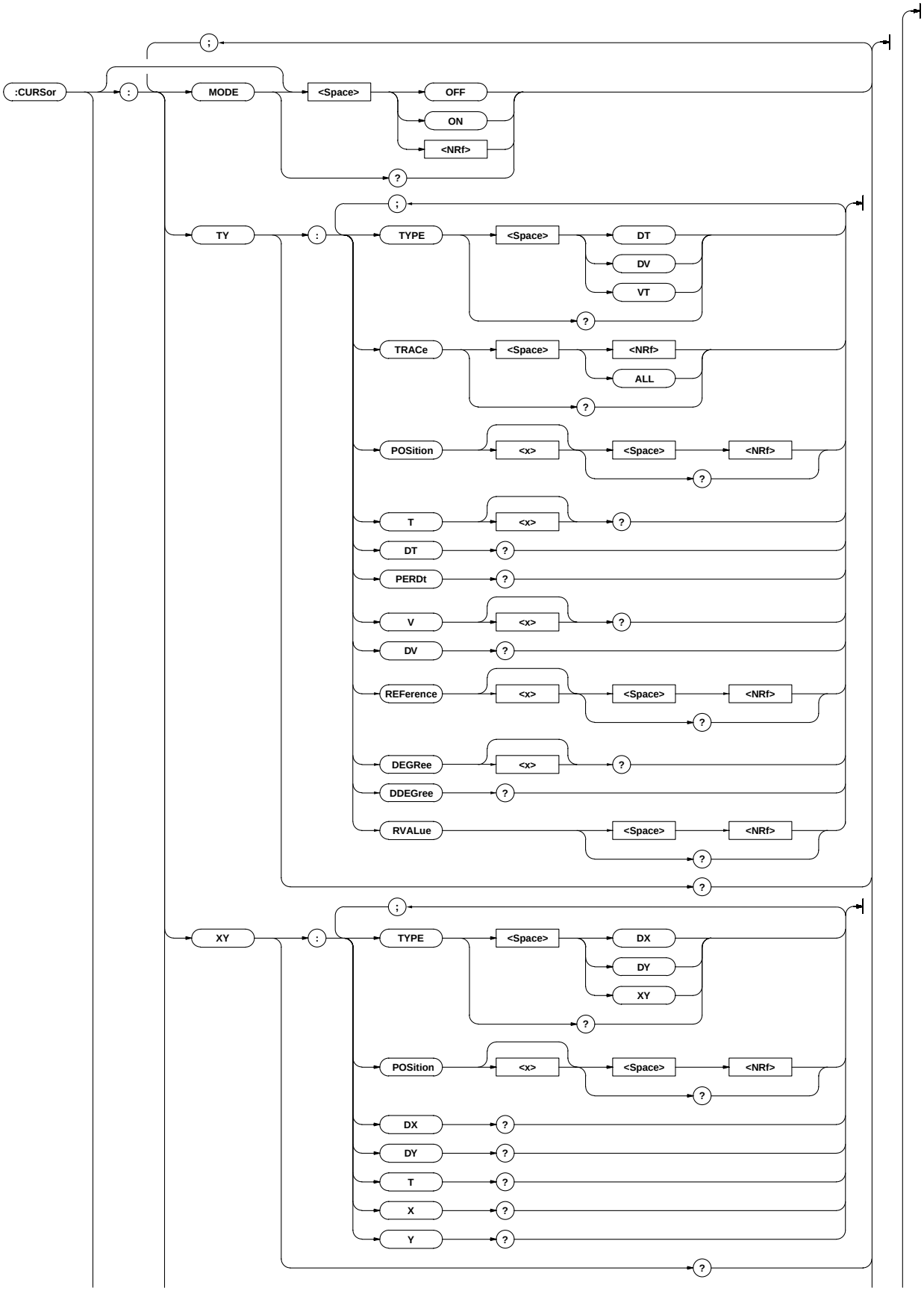
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	MES	0	PRN	0	0	0	0	0

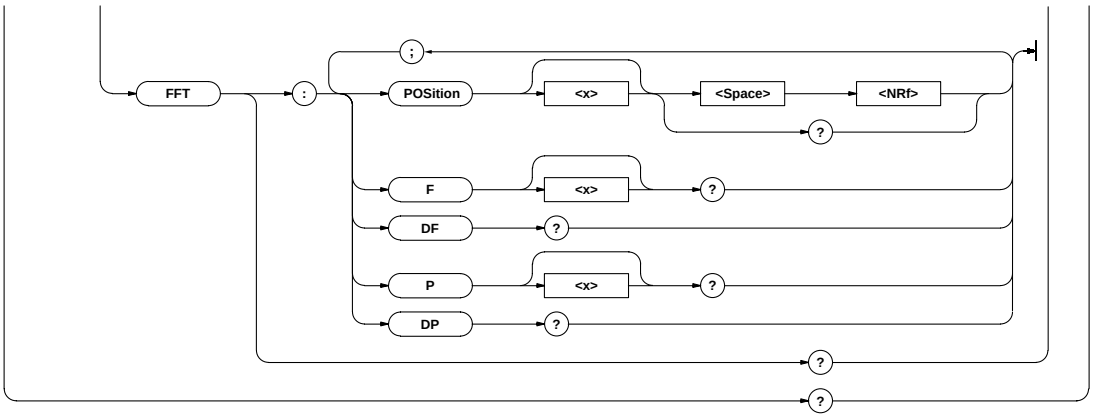
when bit 5(PRN)=1 : printer operation is not completed
when bit 7(MES)=1 : Auto-measurement of ACQ data is not completed

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 inquires 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>}=-|5.00 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:POSiton<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

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?

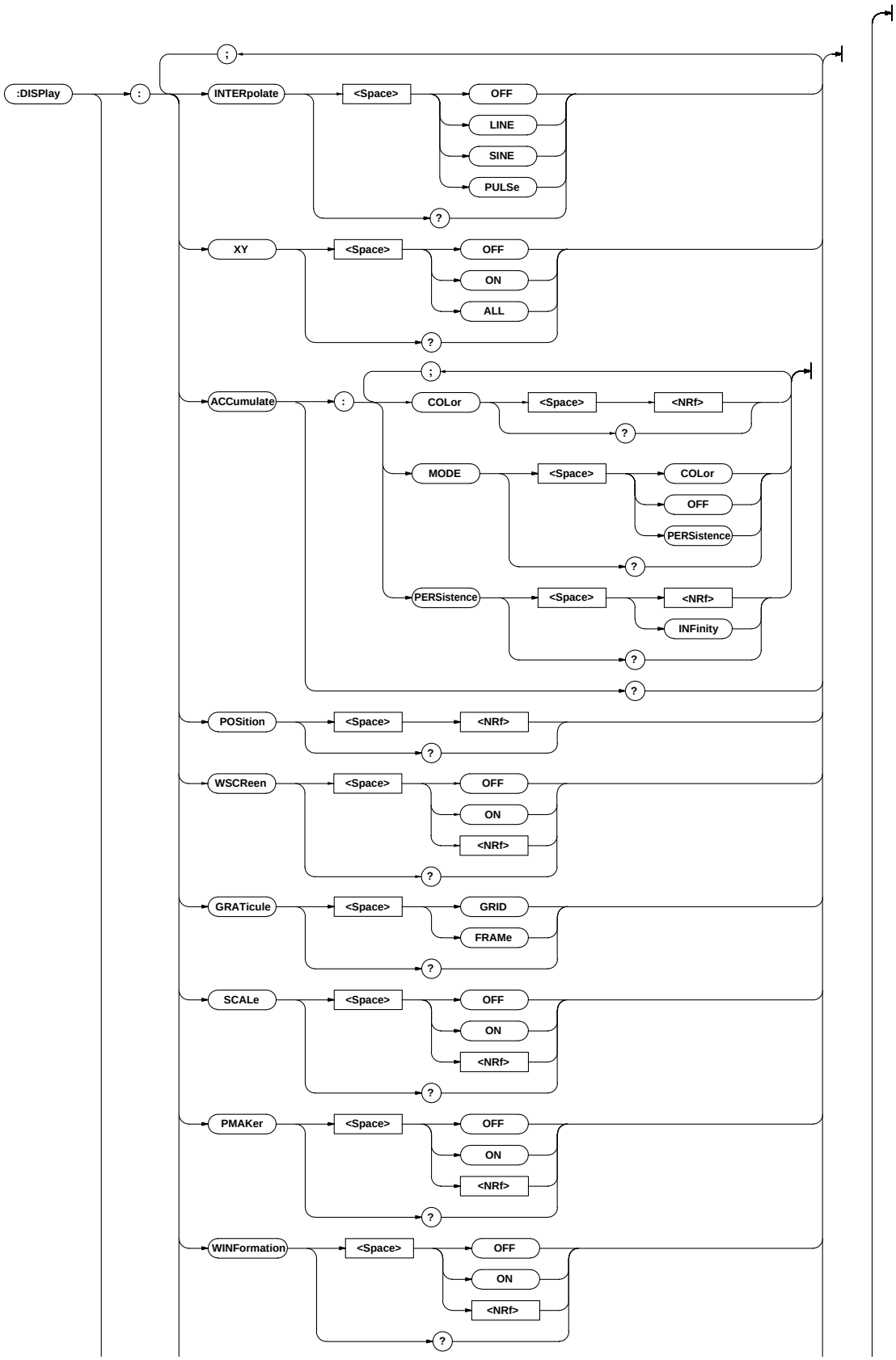
Function Queries the voltage value corresponding to the Y waveform of the X-Y cursor.

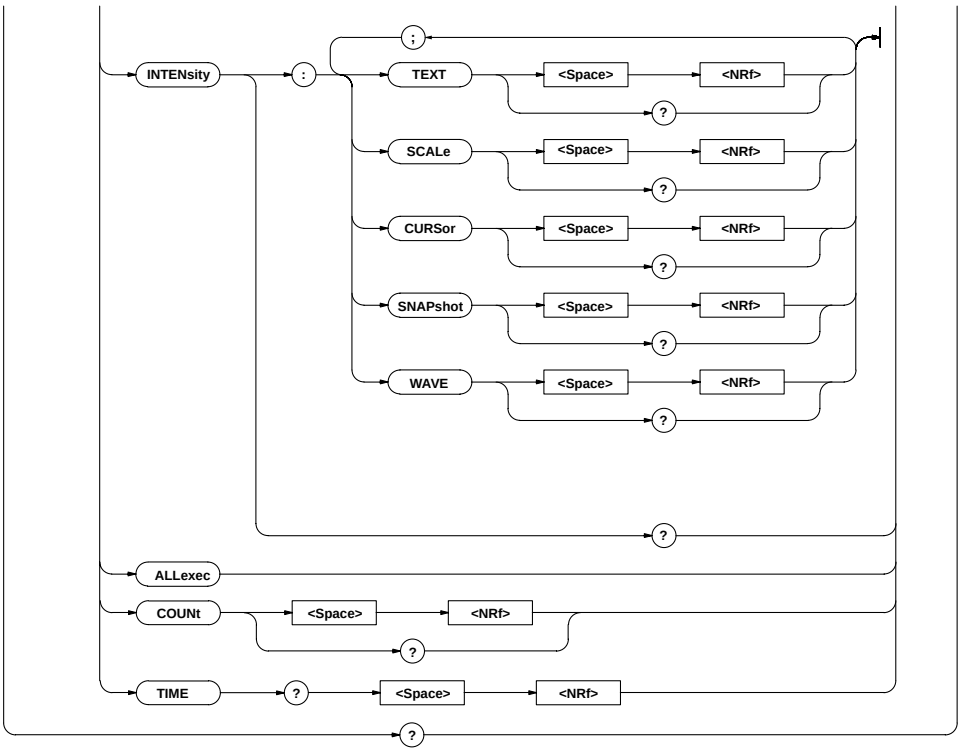
Syntax CURsor:XY:Y?

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^n)

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 0.1S
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

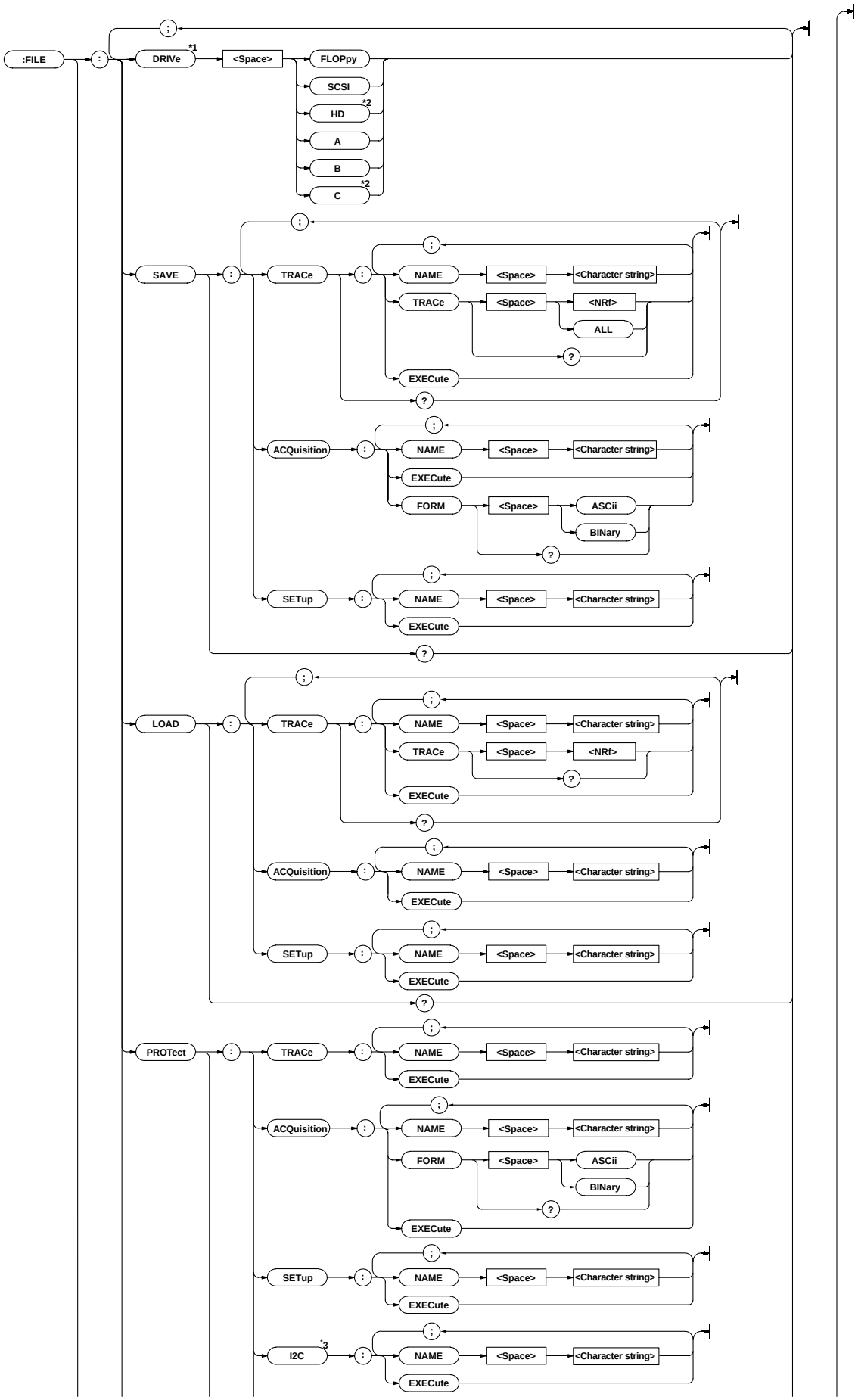
Function Turns the wide screen mode ON/OFF, or queries the current setting.
Syntax DISPlay:WSCReen {<Boolean>}
DISPlay:WSCReen?
Example DISPLAY:WSCREEN ON
DISPLAY:WSCREEN?→:DISPLAY:WSCREEN 1

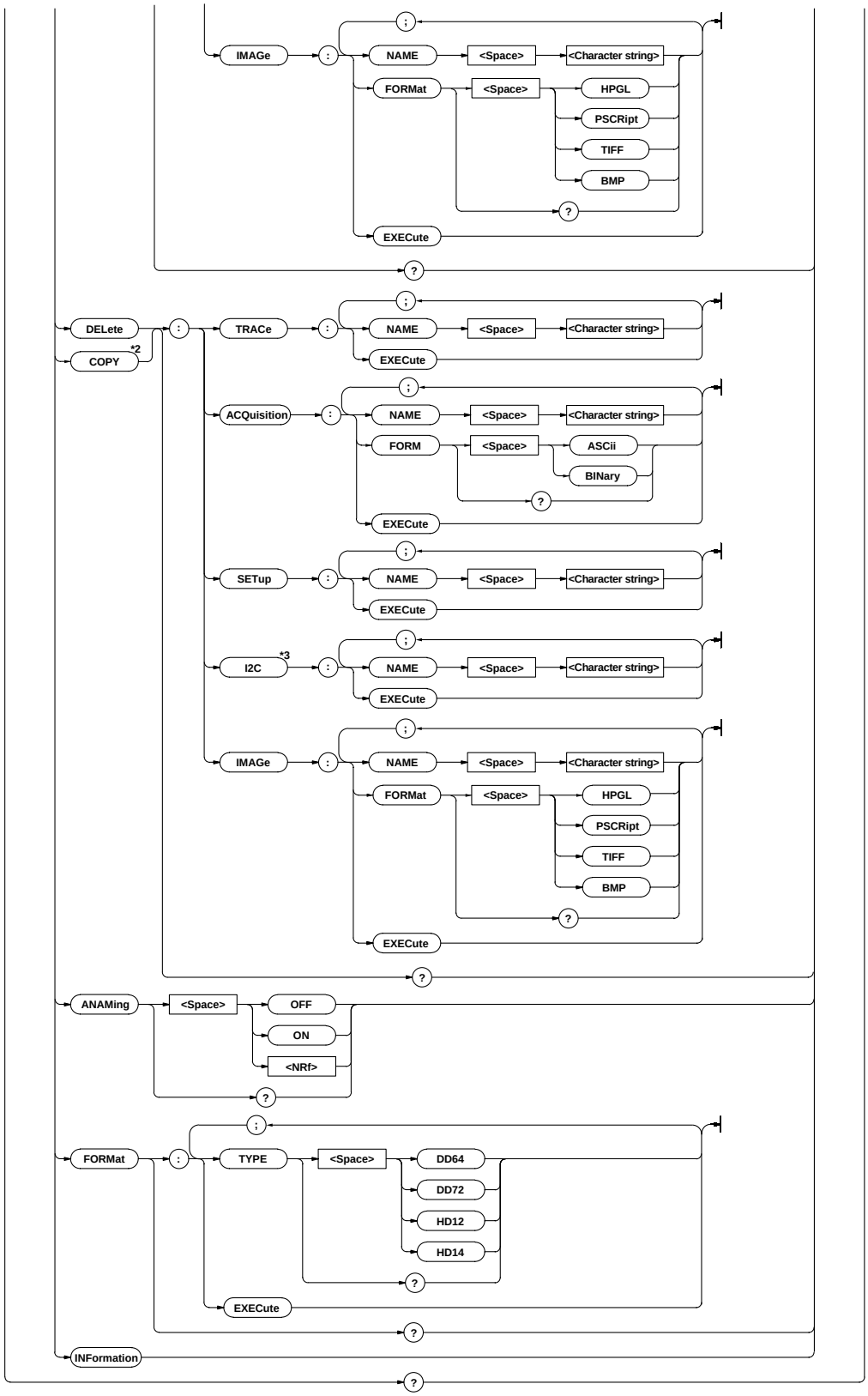
DISPlay:XY

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 on models with the internal hard disk (DL1540CL /C8 option) or through the SCSI interface unit 700930 (sold separately).**
***2 Available only on models with the internal hard disk (DL1540CL /C8 option).**
***3 Available only on models with the I^2C-bus analysis function (/F5 option).**

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:COPY:ACQuisition:EXECute

Function Copies the acquisition data file on the internal hard disk to the floppy disk.
Syntax FILE:COPY:ACQuisition:EXECute
Example FILE:COPY:ACQUISITION:EXECUTE
Description An error occurs if the internal hard disk (option) is not present.

FILE:COPY:ACQuisition:FORM

Function Sets the data format of the waveform to be copied, or queries the current setting.
Syntax FILE:COPY:ACQuisition:FORM {ASCii|BINary}
FILE:COPY:ACQuisition:FORM?
Example FILE:COPY:ACQUISITION:FORM ASCII
FILE:COPY:ACQUISITION:FORM?→:FILE:COPY:ACQUISITION:FORM ASCII
Description An error occurs if the internal hard disk (option) is not present.

FILE:COPY:ACQuisition:NAME

Function Sets the name of the acquisition data file to be copied.
Syntax FILE:COPY:ACQuisition:NAME <character string>
<character string>= up to 8 characters.
See User's Manual IM701530-01E.
Example FILE:COPY:ACQUISITION:NAME "ACQ_1"
Description An error occurs if the internal hard disk (option) is not present.

FILE:COPY:I2C:EXECute

Function Copies the data file containing the I^2C-bus analysis results on the internal hard disk to the floppy disk.
Syntax FILE:COPY:ACQuisition:EXECute
Example FILE:COPY:ACQUISITION:EXECUTE
Description An error occurs if the internal hard disk (option) and the I^2C-bus analysis function (option) are not present.

FILE:COPY:I2C:NAME

Function Sets the name of the file containing the I^2C-bus analysis results to be copied.
Syntax FILE:COPY:I2C:NAME <character string>
<character string>= up to 8 characters.
See User's Manual IM701530-01E.
Example FILE:COPY:I2C:NAME "I2C_1"
Description An error occurs if the internal hard disk (option) and the I^2C-bus analysis function (option) are not present.

FILE:COPY:IMAGe:EXECute

Function Copies the screen image data file on the internal hard disk to the floppy disk.
Syntax FILE:COPY:IMAGe:EXECute
Example FILE:COPY:IMAGE:EXECUTE
Description An error occurs if the internal hard disk (option) is not present.

FILE:COPY:IMAGe:FORMat

Function Sets the data format of the screen image data to be copied, or queries the current setting.
Syntax FILE:COPY:IMAGe:FORMat {HPGL|PSCRipt|TIFF|BMP}
FILE:COPY:IMAGe:FORMat?
Example FILE:COPY:IMAGE:FORMAT HPGL
FILE:COPY:IMAGE:FORMAT?→:FILE:COPY:IMAGE:FORMAT HPGL
Description An error occurs if the internal hard disk (option) is not present.

FILE:COPY:IMAGe:NAME

Function Sets the name of the screen image data file to be copied.
Syntax FILE:COPY:IMAGe:NAME <character string>
<character string>= up to 8 characters.
See User's Manual IM701530-01E.
Example FILE:COPY:IMAGE:NAME "IMAGE_1"
Description An error occurs if the internal hard disk (option) is not present.

FILE:COPY:SETup:EXECute

Function Copies the setting parameter file on the internal hard disk to the floppy disk.
Syntax FILE:COPY:SETup:EXECute
Example FILE:COPY:SETUP:EXECUTE
Description An error occurs if the internal hard disk (option) is not present.

FILE:COPY:SETup:NAME

Function Sets the name of the setting parameter file to be copied.
Syntax FILE:COPY:SETup:NAME <character string>
<character string>= up to 8 characters.
See User's Manual IM701530-01E.
Example FILE:COPY:SETUP:NAME "SETUP_1"
Description An error occurs if the internal hard disk (option) is not present.

FILE:COPY:TRACe:EXECute

Function Copies the display waveform data file on the internal hard disk to the floppy disk.
Syntax FILE:COPY:TRACe:EXECute
Example FILE:COPY:TRACE:EXECUTE
Description An error occurs if the internal hard disk (option) is not present.

FILE:COPY:TRACe:NAME

Function Sets the name of the display waveform data file to be copied.
Syntax FILE:COPY:TRACe:NAME <character string>
<character string>= up to 8 characters.
See User's Manual IM701530-01E.
Example FILE:COPY:TRACE:NAME "TRACE_1"
Description An error occurs if the internal hard disk (option) is not present.

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:FORM

Function Sets the data format of the waveform to be deleted, or queries the current setting.

Syntax
```
FILE:DELete:ACQuisition:FORM {ASCii|
BINary}
FILE:DELete:ACQuisition:FORM?
```

Example
```
FILE:DELETE:ACQUISITION:FORM ASCII
FILE:DELETE:ACQUISITION:FORM?→:FILE:
DELETE:ACQUISITION:FORM ASCII
```

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:I2C:EXECute

Function Deletes the file containing the I^2C-bus analysis results.

Syntax
```
FILE:DELete:I2C:EXECute
```

Example
```
FILE:DELETE:I2C:EXECUTE
```

Description An error occurs if the I^2C-bus analysis function (option) is not present.

FILE:DELete:I2C:NAME

Function Sets the name of the file containing the I^2C-bus analysis results to be deleted.

Syntax
```
FILE:DELete:I2C:NAME <character string>
<character string>= up to 8 characters.
See User's Manual IM701530-01E.
```

Example
```
FILE:DELETE:I2C:NAME "I2C_1"
```

Description An error occurs if the I^2C-bus analysis function (option) is not present.

FILE:DELete:IMAGe:EXECute

Function Deletes the screen image data file.

Syntax
```
FILE:DELete:IMAGe:EXECute
```

Example
```
FILE:DELETE:IMAGE:EXECUTE
```

FILE:DELete:IMAGe:FORMat

Function Sets the data format of the screen image data to be deleted, or queries the current setting.

Syntax
```
FILE:DELete:IMAGe:FORMat {HPGL|PSCRipt|
TIFF|BMP}
FILE:DELete:IMAGe:FORMat?
```

Example
```
FILE:DELETE:IMAGE:FORMAT HPGL
FILE:DELETE:IMAGE:FORMAT?→:FILE:DELETE:
IMAGE:FORMAT HPGL
```

FILE:DELete:IMAGe:NAME

Function Sets the name of the screen image data file to be deleted.

Syntax
```
FILE:DELete:IMAGe:NAME <character string>
<character string>= up to 8 characters.
See User's Manual IM701530-01E.
```

Example
```
FILE:DELETE:IMAGE:NAME "IMAGE_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 storage medium that is to be controlled through the FILE command or queries the current setting.
FLOPpy or A specifies the floppy disk; SCSI or B specifies the SCSI; and HD or C specifies the internal hard disk.
A, B, or C is returned in response to a query.

Syntax
```
FILE:DRIVe {A|B|C|FLOPpy|SCSI|HD}
FILE:DRIVe?
```

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

Description
- SCSI can be used when the internal hard disk (option) is present or through the SCSI interface unit 700930 (sold separately).
- HD can be used when the internal hard disk (option) is present.
- An error occurs on models without the internal hard disk that is not using the SCSI interface unit 700930 (sold separately).

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:FORM

Function Sets the data format of the waveform to be protected, or queries the current setting.

Syntax
```
FILE:PROTect:ACQuisition:FORM {ASCii|
BINary}
FILE:PROTect:ACQuisition:FORM?
```

Example
```
FILE:PROTECT:ACQUISITION:FORM ASCII
FILE:PROTECT:ACQUISITION:FORM?→:FILE:
PROTECT:ACQUISITION:FORM ASCII
```

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:I2C:EXECute

Function Protects the file containing the I^2C-bus analysis results.

Syntax
```
FILE:PROTect:I2C:EXECute
```

Example
```
FILE:PROTect:I2C:EXECUTE
```

Description An error occurs if the I^2C-bus analysis function (option) is not present.

FILE:PROTect:I2C:NAME

Function Sets the name of the file containing the I^2C-bus analysis results to be protected.

Syntax
```
FILE:PROTect:I2C:NAME <character string>
<character string>= up to 8 characters.
See User's Manual IM701530-01E.
```

Example
```
FILE:PROTECT:I2C:NAME "I2C_1"
```

Description An error occurs if the I^2C-bus analysis function (option) is not present.

FILE:PROTect:IMAGe:EXECute

Function Protects the screen image data file.
Syntax FILE:PROTect:IMAGe:EXECute
Example FILE:PROTECT:IMAGE:EXECUTE

FILE:PROTect:IMAGe:FORMat

Function Sets the data format of the screen image data to be protected, or queries the current setting.
Syntax FILE:PROTect:IMAGe:FORMat {HPGL|PSCRipt|TIFF|BMP}
FILE:PROTect:IMAGe:FORMat?
Example FILE:PROTECT:IMAGE:FORMAT HPGL
FILE:PROTECT:IMAGE:FORMAT?→:FILE:PROTECT:IMAGE:FORMAT HPGL

FILE:PROTect:IMAGe:NAME

Function Sets the name of the screen image data file to be protected.
Syntax FILE:PROTect:IMAGe:NAME <character string>
<character string>= up to 8 characters. See User's Manual IM701530-01E.
Example FILE:PROTect:IMAGE:NAME "IMAGE_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: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

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

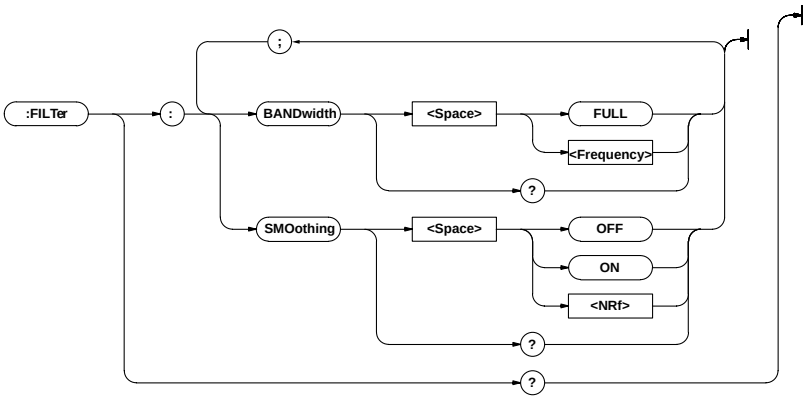
Function Sets the name of the P-P waveform data file to be saved.
Syntax FILE:SAVE:TRACe:NAME <character string>
<character string>= Up to 8 characters can be used, refer to User's manual IM701530-01E.
Example FILE:SAVE:TRACE:NAME "TRACE_1"

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

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 inquires 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

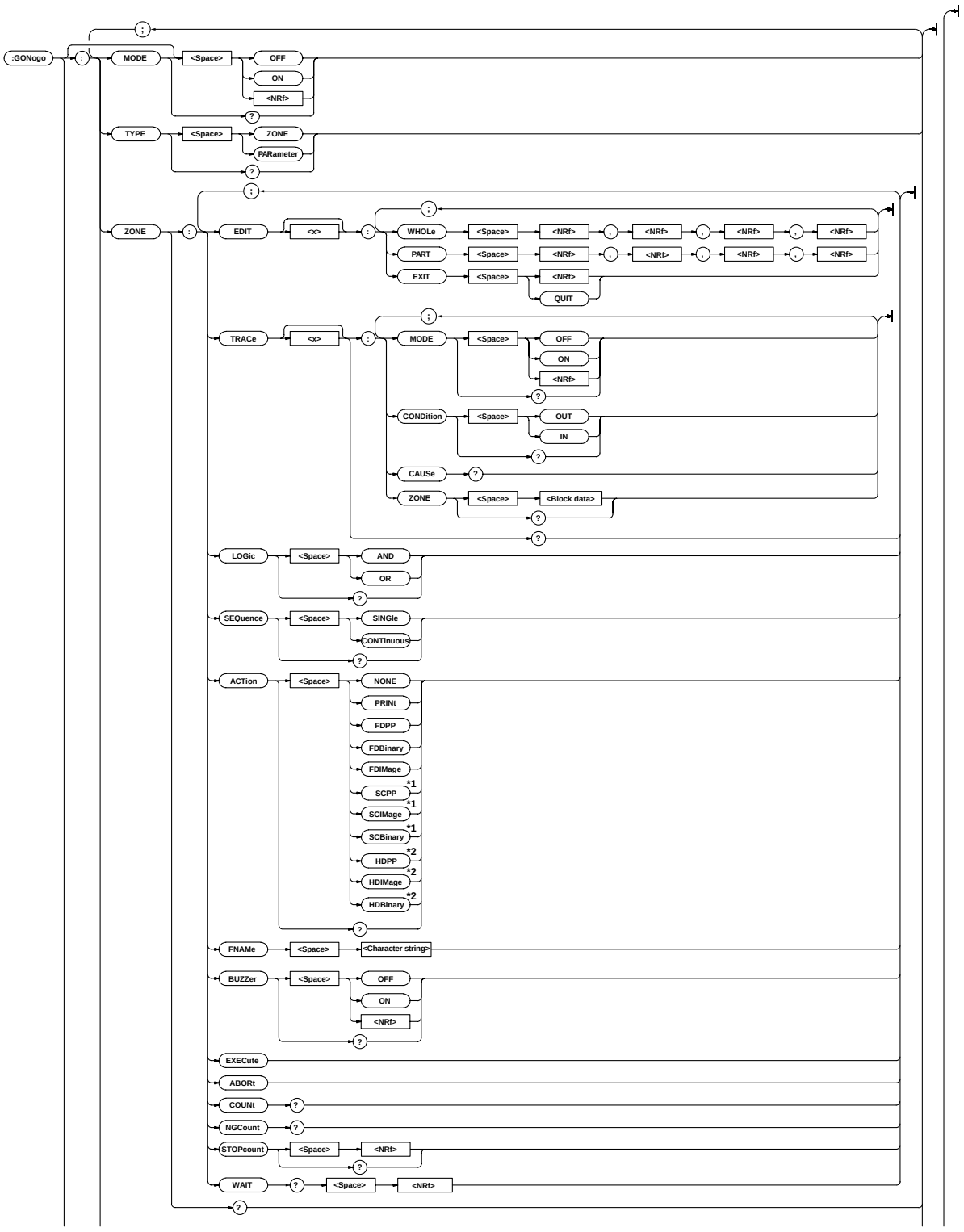
Function Turns smoothing ON/OFF, or queries about the current setting.

Syntax
```
FILTer:SMOothing {<Boolean>}
FILTer:SMOothing?
```

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

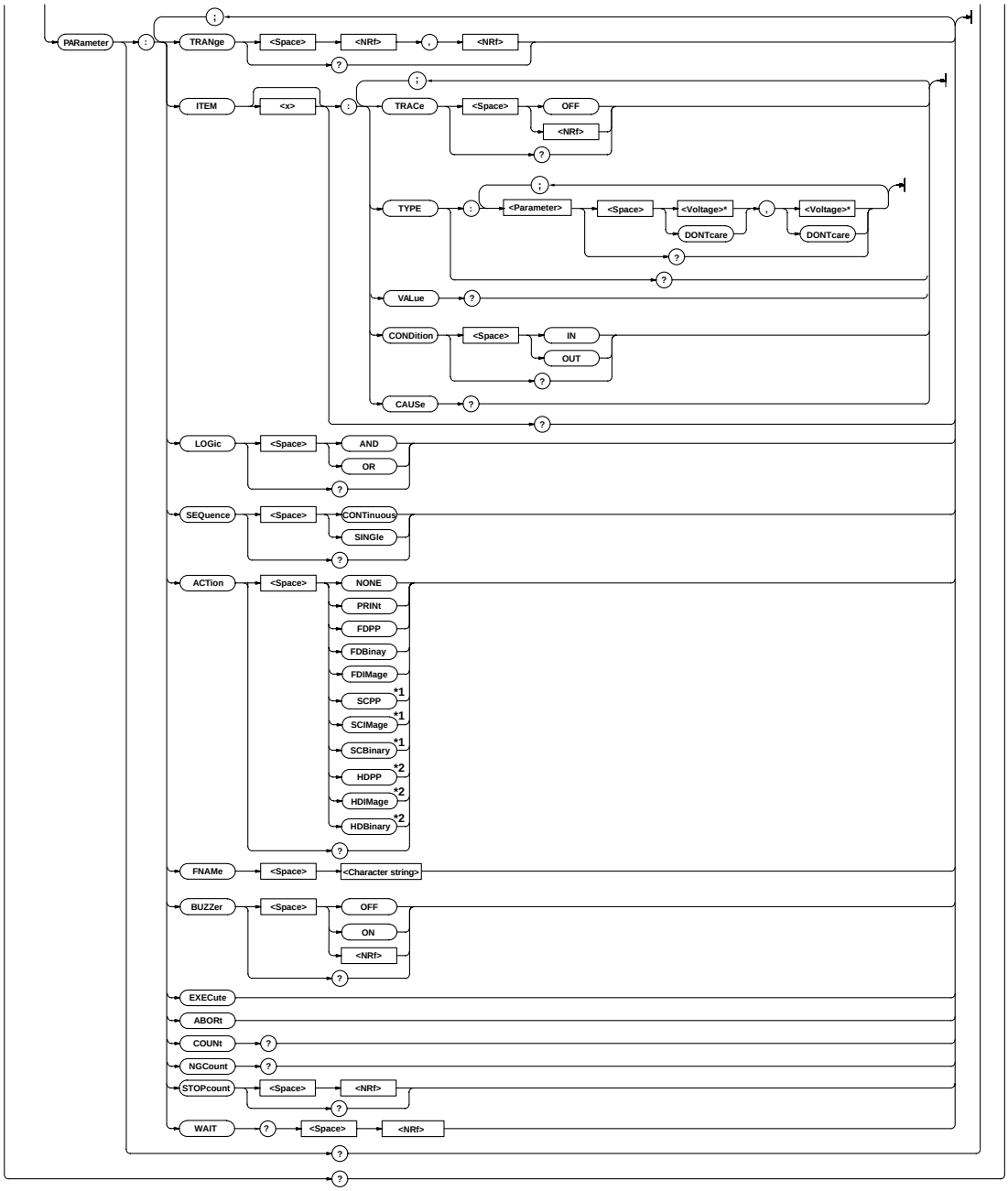
4.12 GONogo Group

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 on models with the internal hard disk (DL1540CL /C8 option) or through the SCSI interface unit 700930 (sold separately).**
***2 Available only on models with the internal hard disk (DL1540CL /C8 option).**

* 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



***1 Available on models with the internal hard disk (DL1540CL /C8 option) or through the SCSI interface unit 700930 (sold separately).**
***2 Available only on models with the internal hard disk (DL1540CL /C8 option).**

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>}
GONogo:MODE?
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|HDPP|HDBinary|HDIMage}
GONogo:PARameter:ACTion?
Example GONOGO:PARAMETER:ACTION NONE
GONOGO:PARAMETER:ACTION?→:GONOGO:PARAMETER:ACTION NONE
Description
- SCPP, SCBinary, and SCIMage can be used through the SCSI interface unit 700930 (sold separately) or the SCSI interface connector (/C8 option).
- SCPP, SCBinary and SCIMage are available only when SCSI is connected.
- HDPP, HDBinary, and HDIMage are available only when the internal hard disk (option) is present.

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>}
GONogo:PARameter:BUZZer?
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 quiries 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|
 XY1Int eg|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|HDPP|HDBinary|HDIMage}
GONogo:ZONE:ACTion?
```

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

Description
- SCPP, SCBinary, and SCIMage can be used through the SCSI interface unit 700930 (sold separately) or the SCSI interface connector (/C8 option).
- SCPP, SCBinary and SCIMage are available only when SCSI is connected.
- HDPP, HDBinary, and HDIMage are available only when the internal hard disk (option) is present.

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.

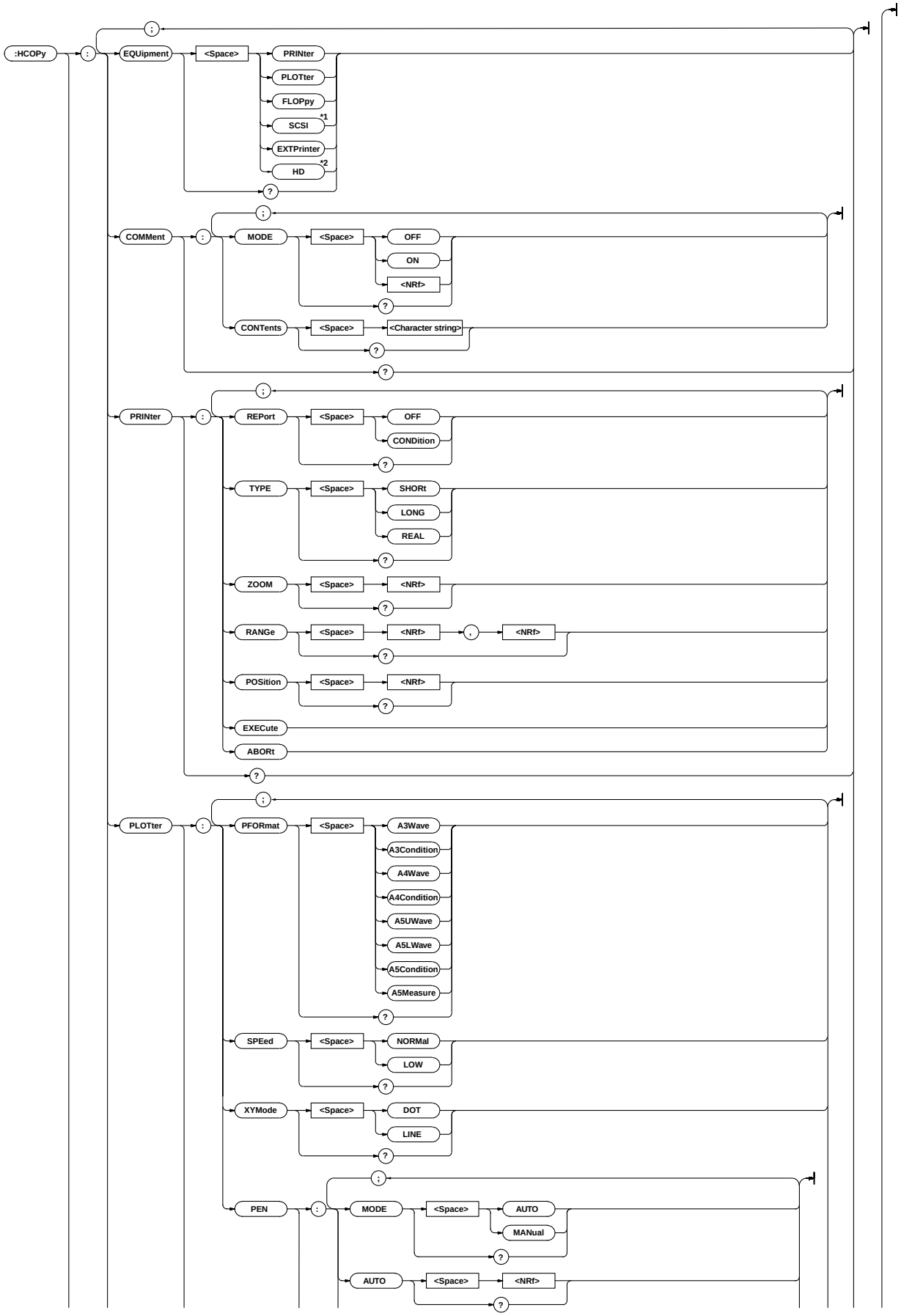
Syntax
```
GONogo:ZONE:WAIT? {<NRf>}
{<NRf>}=0 to 864000 (in units of 100 ms)
```

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

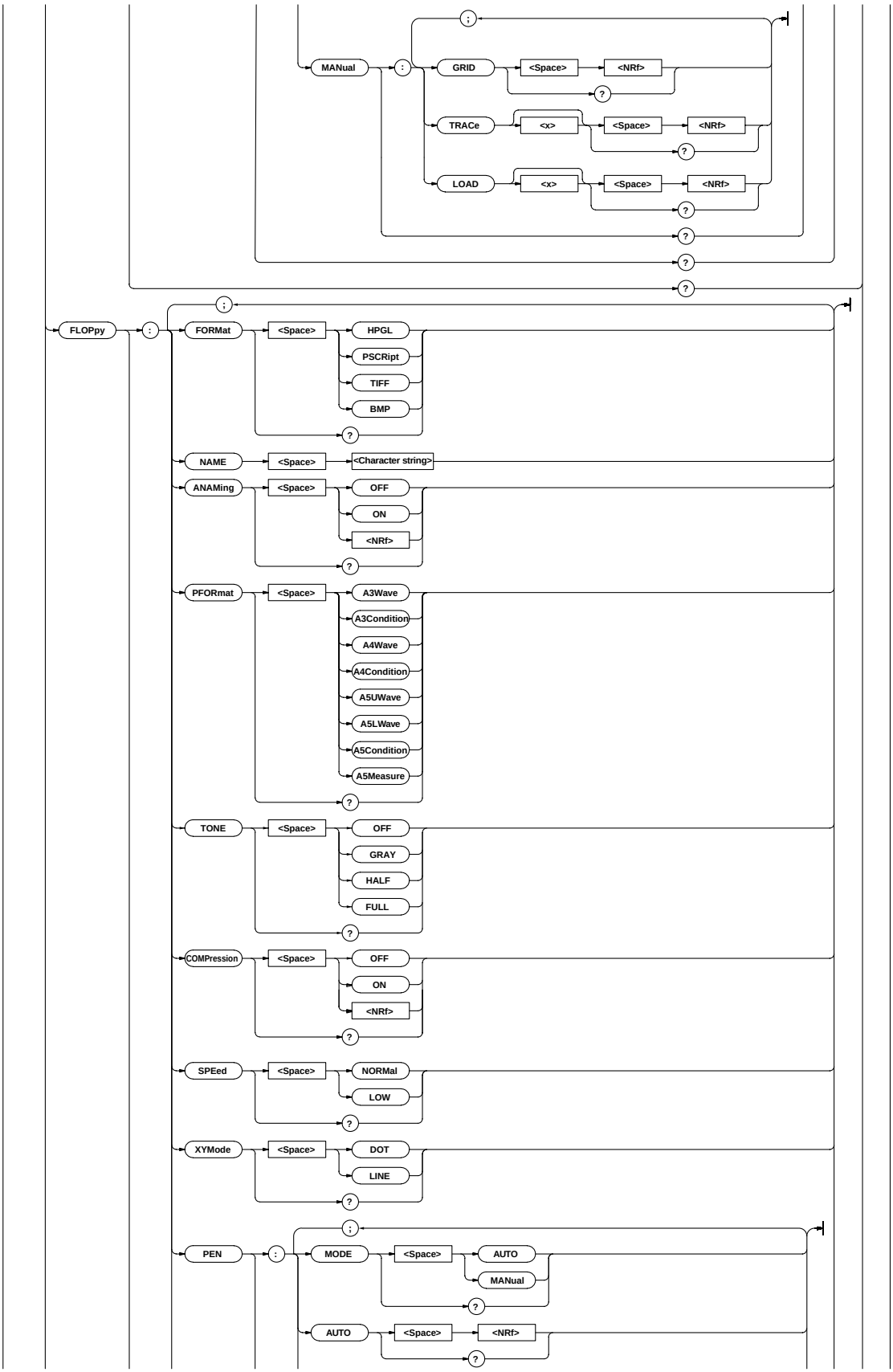
Description “1” is returned if it timed out: “0” if finished in time.

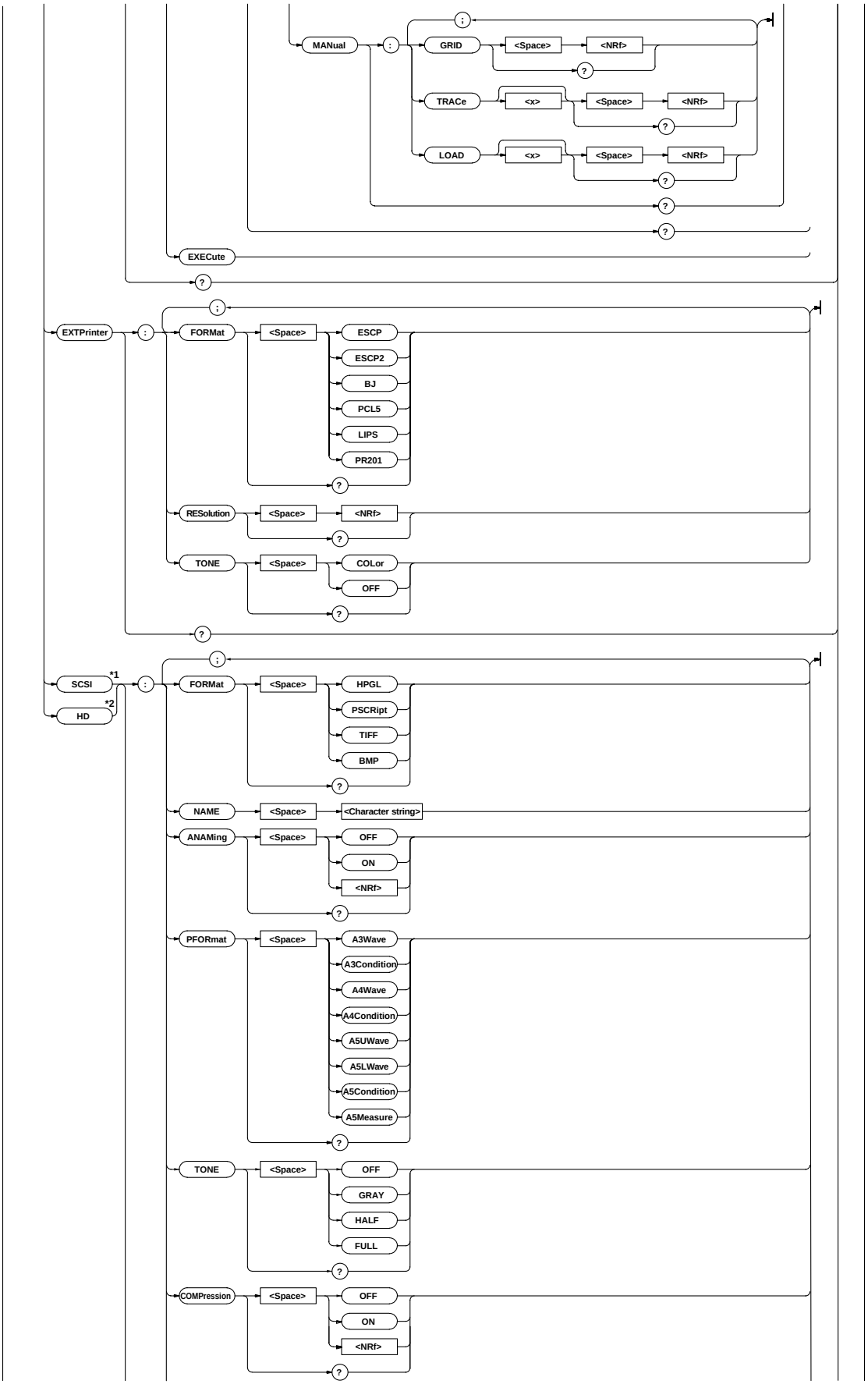
4.13 HCOPy Group

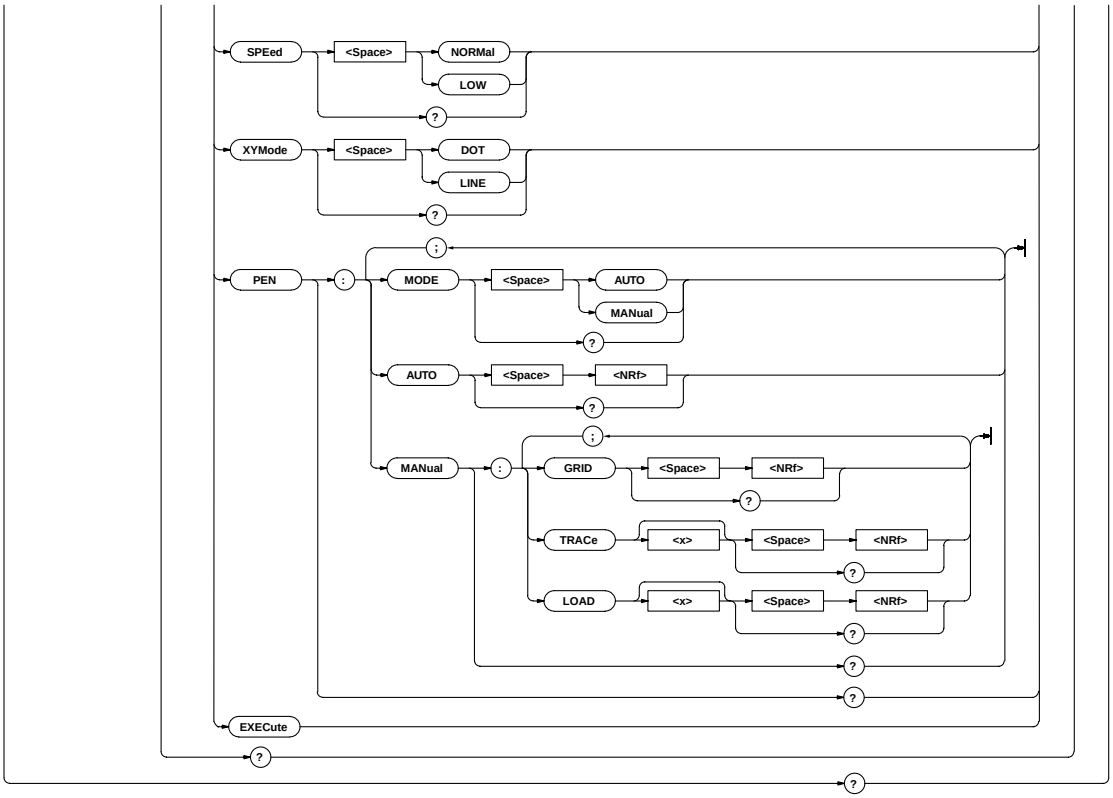
The commands in the HCOPy group are used to make settings relating to, and inquiries about screen data output to the optional built-in printer or external HP-GL plotter. You can make the same settings and inquiries as when using the COPY key on the front panel.



*1 Available on models with the internal hard disk (DL1540CL /C8 option) or through the SCSI interface unit 700930 (sold separately).
*2 Available only on models with the internal hard disk (DL1540CL /C8 option).


;
MANual
:
GRID
<Space>
<NRf>
?
TRACe
<x>
<Space>
<NRf>
?
LOAD
<x>
<Space>
<NRf>
?
?
?
?
;
FLOPpy
:
FORMat
<Space>
HPGL
PSCRipt
TIFF
BMP
?
NAME
<Space>
<Character string>
ANAMing
<Space>
OFF
ON
<NRf>
?
PFORmat
<Space>
A3Wave
A3Condition
A4Wave
A4Condition
A5UWave
A5LWave
A5Condition
A5Measure
?
TONE
<Space>
OFF
GRAY
HALF
FULL
?
COMPression
<Space>
OFF
ON
<NRf>
?
SPEed
<Space>
NORMal
LOW
?
XYMode
<Space>
DOT
LINE
?
;
PEN
:
MODE
<Space>
AUTO
MANual
?
AUTO
<Space>
<NRf>
?


MANual
GRID
<Space>
<NRf>
TRACe
<x>
LOAD
EXECute
EXTPrinter
FORMat
ESCP
ESCP2
BJ
PCL5
LIPS
PR201
RESolution
TONE
COLor
OFF
SCSI
*1
HD
*2
HPGL
PSCRipt
TIFF
BMP
NAME
<Character string>
ANAMing
ON
PFORmat
A3Wave
A3Condition
A4Wave
A4Condition
A5UWave
A5LWave
A5Condition
A5Measure
GRAY
HALF
FULL
COMPression



***1 Available on models with the internal hard disk (DL1540CL /C8 option) or through the SCSI interface unit 700930 (sold separately).**
***2 Available only on models with the internal hard disk (DL1540CL /C8 option).**

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|HD|SCSI}
HCOPy:EQUipment?
```

Example
```
HCOPY:EQUIPMENT PRINTER
HCOPY:EQUIPMENT?→:HCOPY:
EQUIPMENT PRINTER
```

Description
- The SCSI can be used through the SCSI interface unit 700930 (sold separately) or the SCSI interface connector (/C8 option).
- The SCSI is available only when it is connected.
- HD can be used when the internal hard disk (option) is present.

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:RESolution

Function Sets the output resolution when printing in BJ format, or queries the current setting.

Syntax
```
HCOPy:EXTPrinter:RESolution {<NRf>}
HCOPy:EXTPrinter:RESolution?
<NRf> = 180,300,360(dpi)
```

Example
```
HCOPY:EXTPRINTER:RESOLUTION 360
HCOPY:EXTPRINTER:RESOLUTION?→:HCOPY:
EXTPRINTER:RESOLUTION 360
```

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 {<Boolean>}
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 {<Boolean>}
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:MDOE 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 wether sampling points are connected by dots (DOT) or lines (LINE).

HCOPy:HD?

Function Queries the setting parameters related to saving of the screen image data to the internal hard disk.

Syntax HCOPy:HD?

Example HCOPy:HD?→:HCOPY:HD: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
- An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:ANAMing

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

Syntax HCOPy:HD:ANAMing {<Boolean>}
HCOPy:HD:ANAMing?

Example HCOPY:HD:ANAMING ON
HCOPY:HD:ANAMING?→:HCOPY:HD:ANAMING ON

Description An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:COMPression

Function Turns ON/OFF the compression for the screen image data output, or queries the current setting.

Syntax HCOPy:HD:COMPression {<Boolean>}
HCOPy:HD:COMPression?

Example HCOPY:HD:COMPRESSION ON
HCOPY:HD:COMPRESSION?→:HCOPY:HD:COMPRESSION 1

Description An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:EXECute

Function Saves the screen image data to the internal hard disk.

Syntax HCOPy:HD:EXECute

Example HCOPY:HD:EXECUTE

Description An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:FORMat

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

Syntax HCOPy:HD:FORMat {HPGL|PSCRipt|TIFF|BMP}
HCOPy:HD:FORMat?

Example HCOPY:HD:FORMAT TIFF
HCOPY:HD:FORMAT?→:HCOPY:HD:FORMAT TIFF

Description An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:NAME

Function Selects the name of the screen image data file to be output to the internal hard disk.

Syntax HCOPy:HD:NAME {<character string>}
<character string>= up to 8 characters

Example HCOPY:HD: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.
- An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:PEN?

Function Queries all the settings related to the pens used to output the image data to the internal hard disk.

Syntax HCOPy:HD:PEN?

Example HCOPY:HD:PEN?→:HCOPY:HD: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 An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:PEN:AUTO

Function Sets the number of pens when the pens used to output the screen image data to the internal hard disk are assigned automatically, or queries the current setting.

Syntax HCOPy:HD:PEN:AUTO {<NRf>}
HCOPy:HD:PEN:AUTO?
<NRf>=1 to 5

Example HCOPY:HD:PEN:AUTO 5
HCOPY:HD:PEN:AUTO?→:HCOPY:HD:PEN:AUTO 5

Description An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:PEN:MANual?

Function Queries all settings when the pens used to output the screen image data to the internal hard disk are assigned manually.

Syntax HCOPy:HD:PEN:MANual?

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

Description An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:PEN:MANual:GRID

Function Sets the pen numbers for the graticule, cursor, etc. when the pens used to output the screen image data to the internal hard disk are assigned manually, or queries the current setting.

Syntax HCOPy:HD:PEN:MANual:GRID {<NRf>}
HCOPy:HD:PEN:MANual:GRID?
<NRf>=0 to 12

Example HCOPY:HD:PEN:MANUAL:GRID 1
HCOPY:HD:PEN:MANUAL:GRID?→:HCOPY:HD:PEN:MANUAL:GRID 1

Description An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:PEN:MANual:LOAD<x>

Function Sets the pen number for each loaded trace when the pens used to output the screen image data to the internal hard disk are assigned manually, or queries the current setting.

Syntax
```
HCOPy:HD:PEN:MANual:LOAD<x> {<NRf>}
HCOPy:HD:PEN:MANual:LOAD<x>?
<x>=1 to 4
<NRf>=0 to 12
```

Example
```
HCOPY:HD:PEN:MANUAL:LOAD1 6
HCOPY:HD:PEN:MANUAL:LOAD1?→:HCOPY:HD:
PEN:MANUAL:LOAD1 6
```

Description An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:PEN:MANual:TRACe<x>

Function Sets the pen number for each trace when the pens used to output the screen image data to the internal hard disk are assigned manually, or queries the current setting.

Syntax
```
HCOPy:HD:PEN:MANual:TRACe<x> {<NRf>}
HCOPy:HD:PEN:MANual:TRACe<x>?
<x>=1 to 4
<NRf>=0 to 12
```

Example
```
HCOPY:HD:PEN:MANUAL:TRACE1 2
HCOPY:HD:PEN:MANUAL:TRACE1?→:HCOPY:HD:
PEN:MANUAL:TRACE1 2
```

Description An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:PEN:MODE

Function Sets the pen assignment method for the screen image data to be output to the internal hard disk, or queries the current setting.

Syntax
```
HCOPy:HD:PEN:MODE {AUTO|MANual}
HCOPy:HD:PEN:MODE?
```

Example
```
HCOPY:HD:PEN:MODE AUTO
HCOPY:HD:PEN:MODE?→:HCOPY:HD:PEN:
MODE AUTO
```

Description An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:PFORmat

Function Sets the size of the screen image data to be output to the internal hard disk, or queries the current setting.

Syntax
```
HCOPy:HD:PFORmat {A3Wave|A3Condition|
A4Wave|A4Condition|A5UWave|A5LWave|
A5Condition|A5Measure}
HCOPy:HD:PFORmat?
```

Example
```
HCOPY:HD:PFORMAT A4WAVE
HCOPY:HD:PFORMAT?→:HCOPY:HD:
PFORMAT A4WAVE
```

Description An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:SPEed

Function Sets the pen speed used to output the screen image data to the internal hard disk, or queries the current setting.

Syntax
```
HCOPy:HD:SPEed {NORMal|LOW}
HCOPy:HD:SPEed?
```

Example
```
HCOPY:HD:SPEED NORMAL
HCOPY:HD:SPEED?→:HCOPY:HD:SPEED NORMAL
```

Description An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:TONE

Function Sets the color tone of the screen image data output, or queries the current setting.

Syntax
```
HCOPy:HD:TONE {OFF|GRAY|HALF|FULL}
HCOPy:HD:TONE?
```

Example
```
HCOPY:HD:TONE FULL
HCOPY:HD:TONE?→:HCOPY:HD:TONE FULL
```

Description An error occurs if the internal hard disk (option) is not present.

HCOPy:HD:XYMode

Function Selects the plot method for X-Y waveforms that are to be output to the internal hard disk, or queries the current setting.

Syntax
```
HCOPy:HD:XYMode {DOT|LINE}
HCOPy:HD:XYMode?
```

Example
```
HCOPY:HD:XYMODE LINE
HCOPY:HD:XYMODE?→:HCOPY:HD:XYMODE LINE
```

Description
- You can select dots (DOT) or lines (LINE) for connecting the sampled points.
- An error occurs if the internal hard disk (option) is not present.

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:MDOE 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|
A5LWave|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 wether 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 ouputting 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 <charater 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.

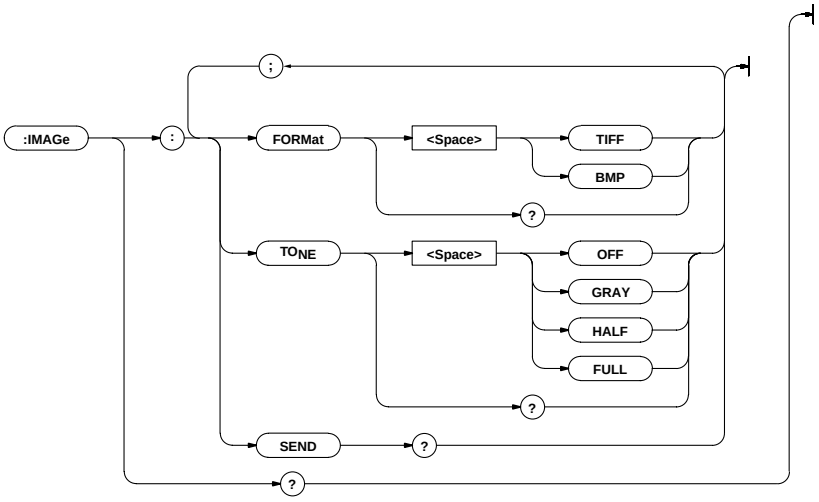
Syntax
```
HCOPy:SCSI:XYMode {DOT|LINE}
HCOPy:SCSI:XYMode?
```

Example
```
HCOPY:SCSI:XYMODE LINE
HCOPY:SCSI:XYMODE?→:HCOPY:SCSI:
XYMODE LINE
```

Description
- It can be selected irrespective of wether 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+{Number of data+{1(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

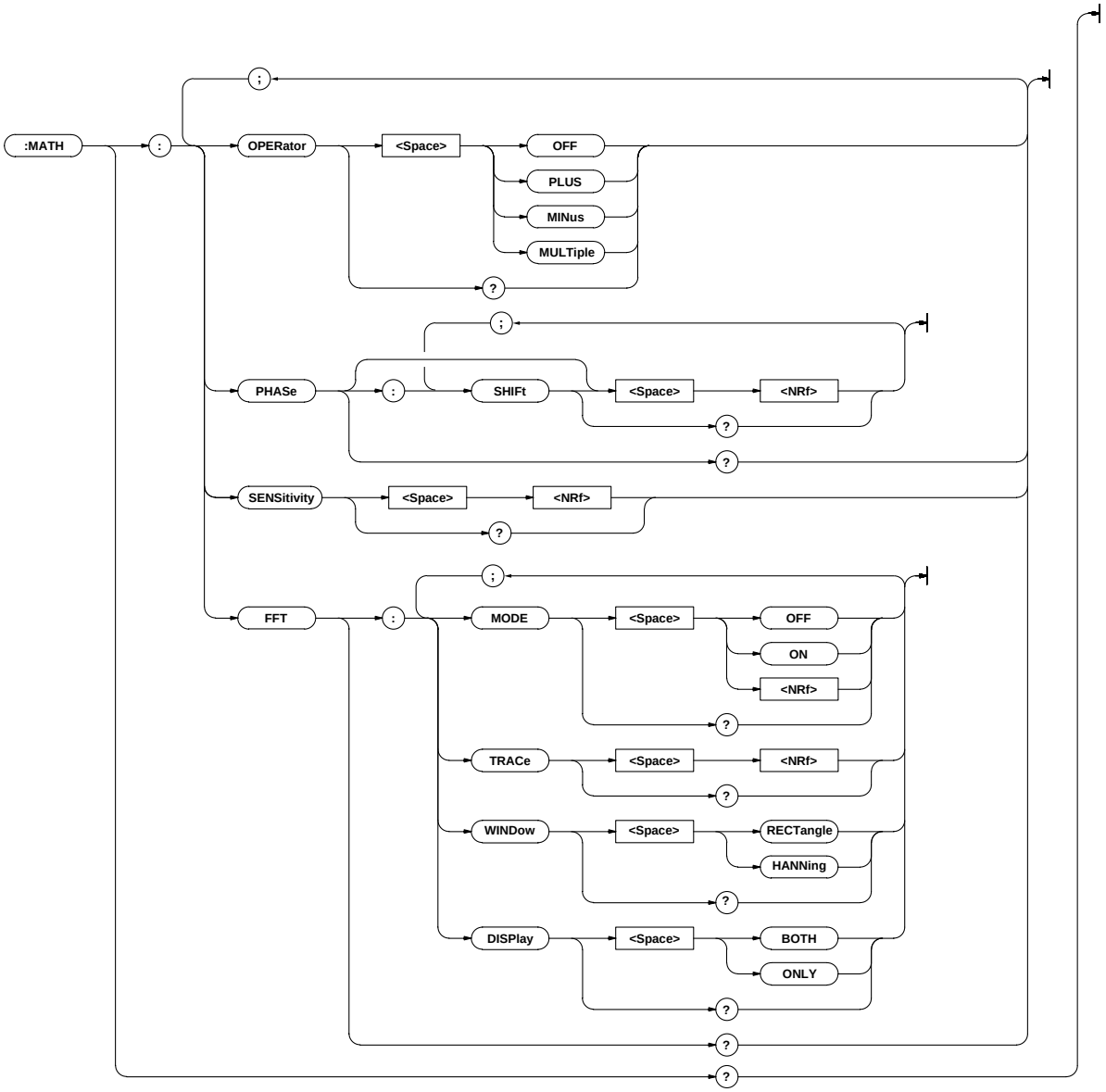
The INITialize command is used for making settings relating to initialization. This allows you to make the same settings as when using the INITIALIZE key on the front panel.

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

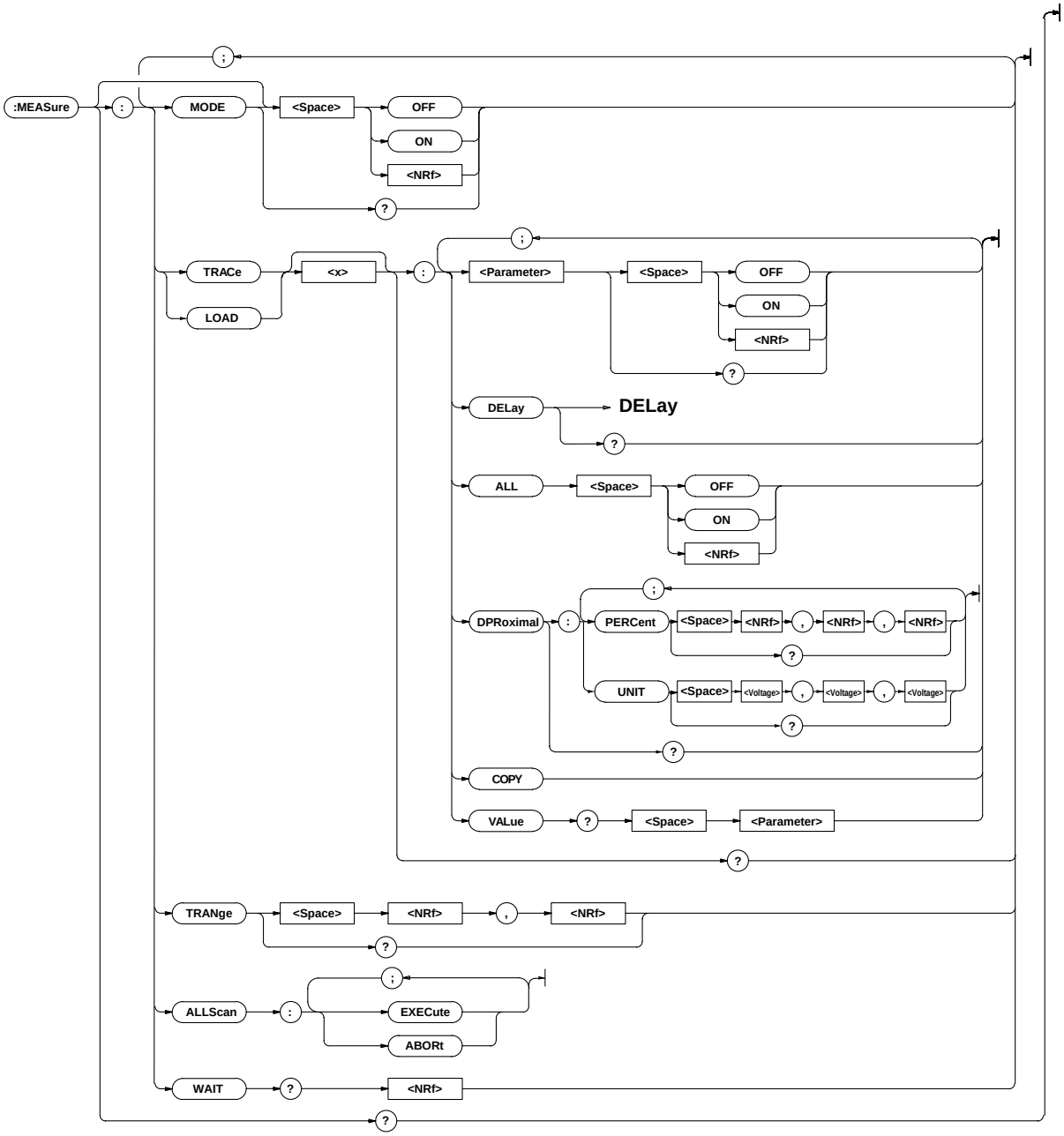
Function Sets the display sensititivity of the vertical axis, or queries the current setting.

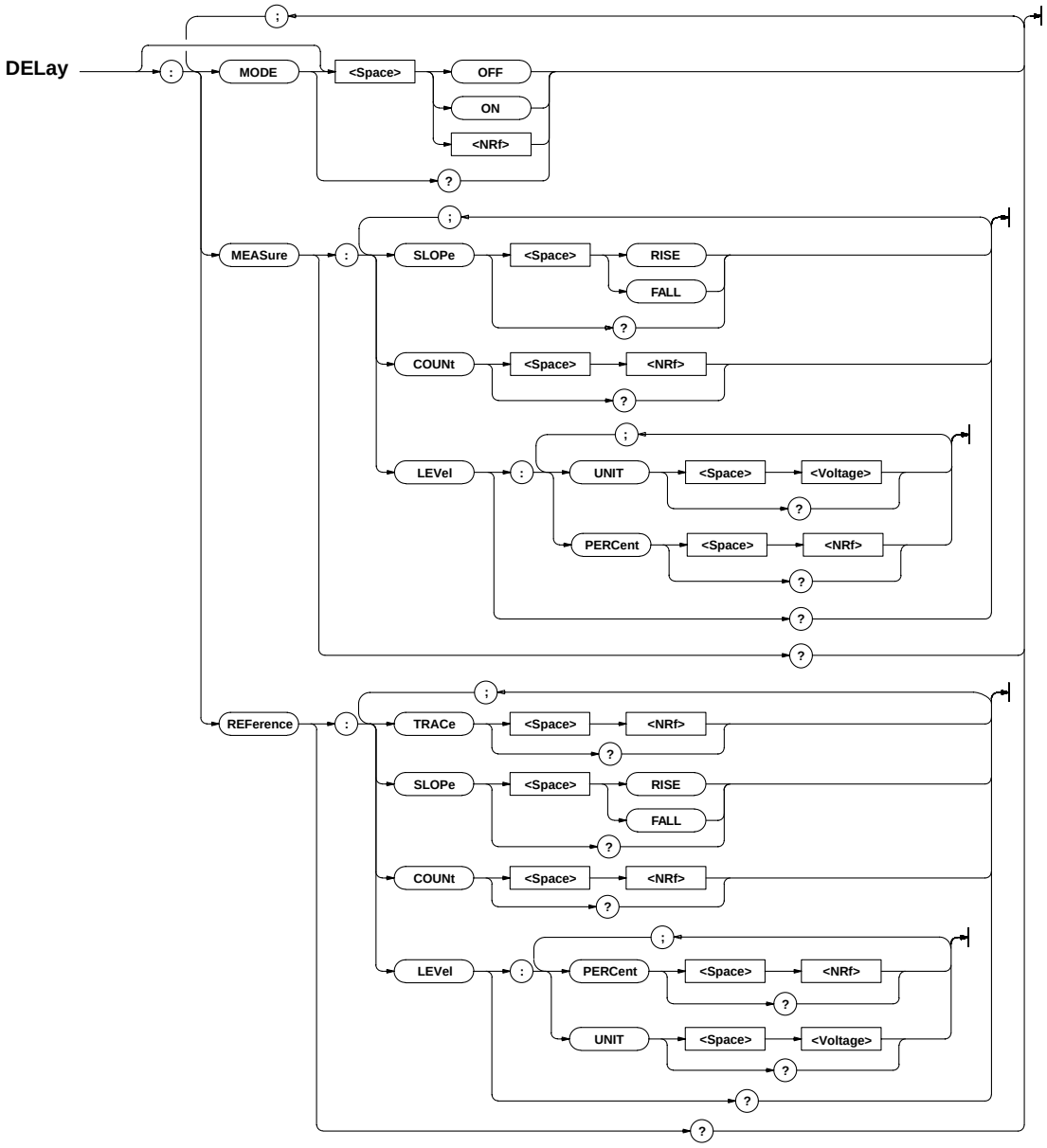
Syntax
```
MATH:SENSitivity {<NRf>}
MATH:SENSitivity?
{<NRf>}=0.50 to 4.00 (div, 0.01 step)
```

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.




DELay
:
;
MODE
<Space>
OFF
ON
<NRf>
?
MEASure
SLOPe
RISE
FALL
COUNt
LEVel
UNIT
<Voltage>
PERCent
REFerence
TRACe

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 treshold 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 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: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 beween 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|BWIDth|DUTYcycle|
              FALL|FREQuency|HIGH|LOW|
              MAXimum|MINimum|NOVershoot|
              NWIDth|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.

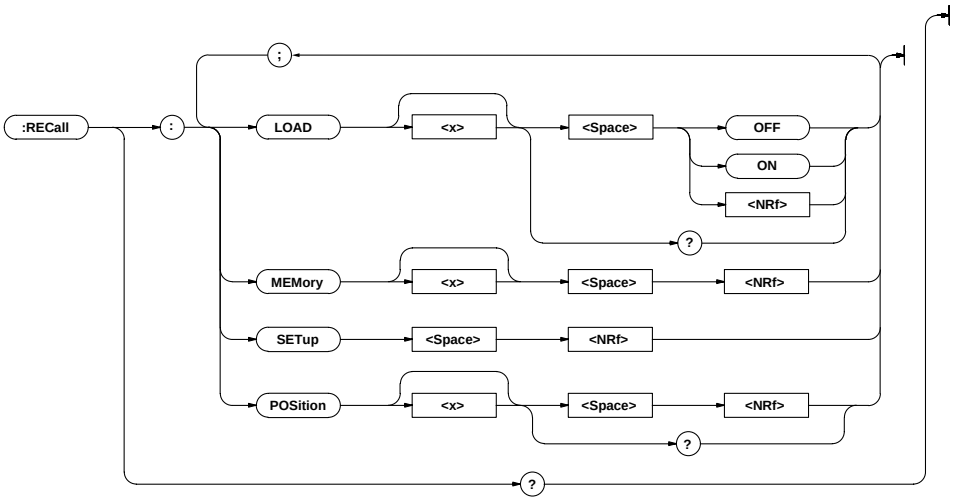
Syntax
```
MEASure:WAIT? {<NRf>}
{<NRf>}=1 to 36000 (time-out period, in
        units of 100 ms)
```

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

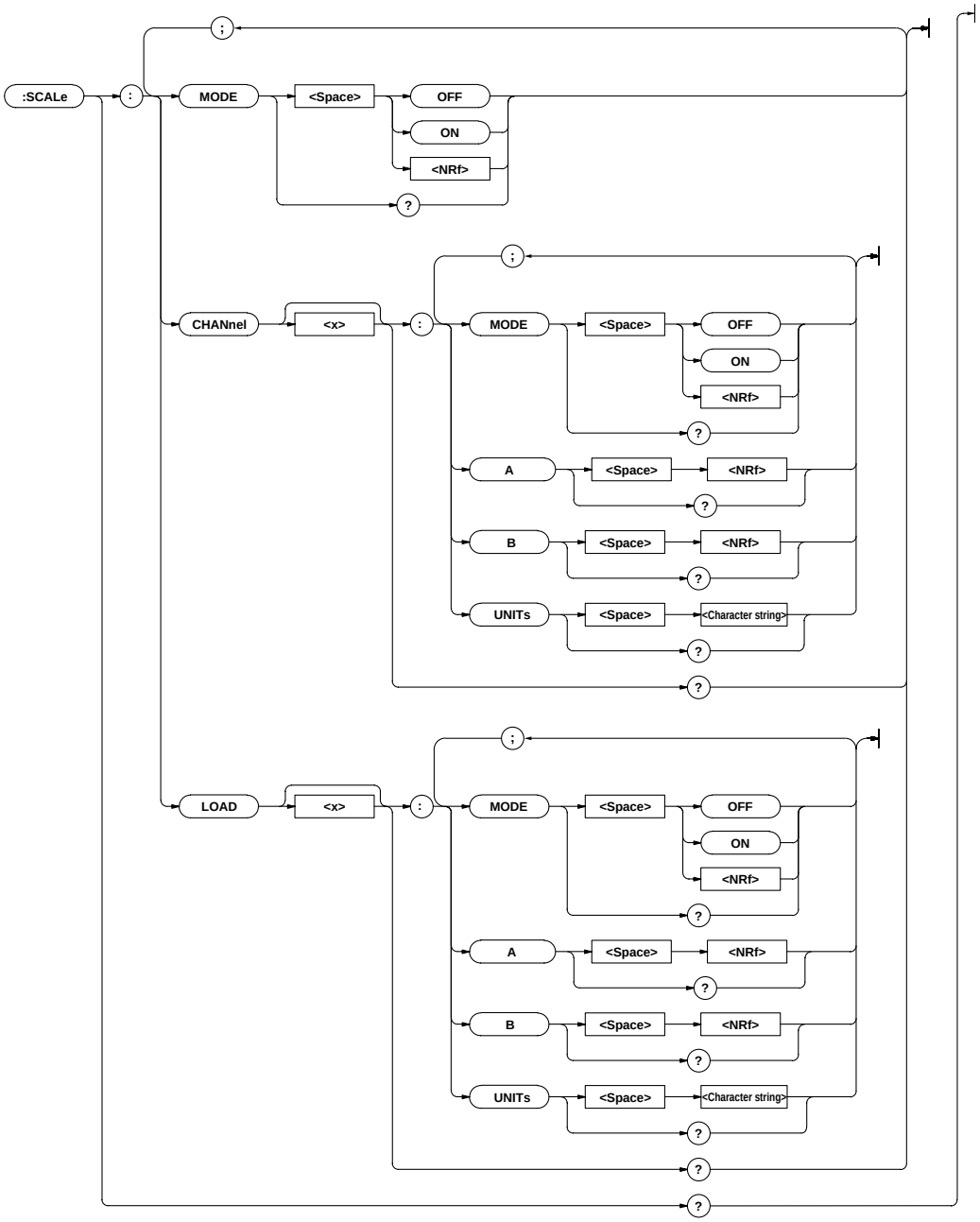
Function Recalls the specified set-up data from the internal mememory.

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 A 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

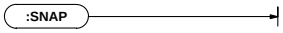
Function Sets linear scaling ON/OFF, or queries the current setting.

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.

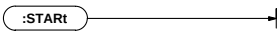


SNAP

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.

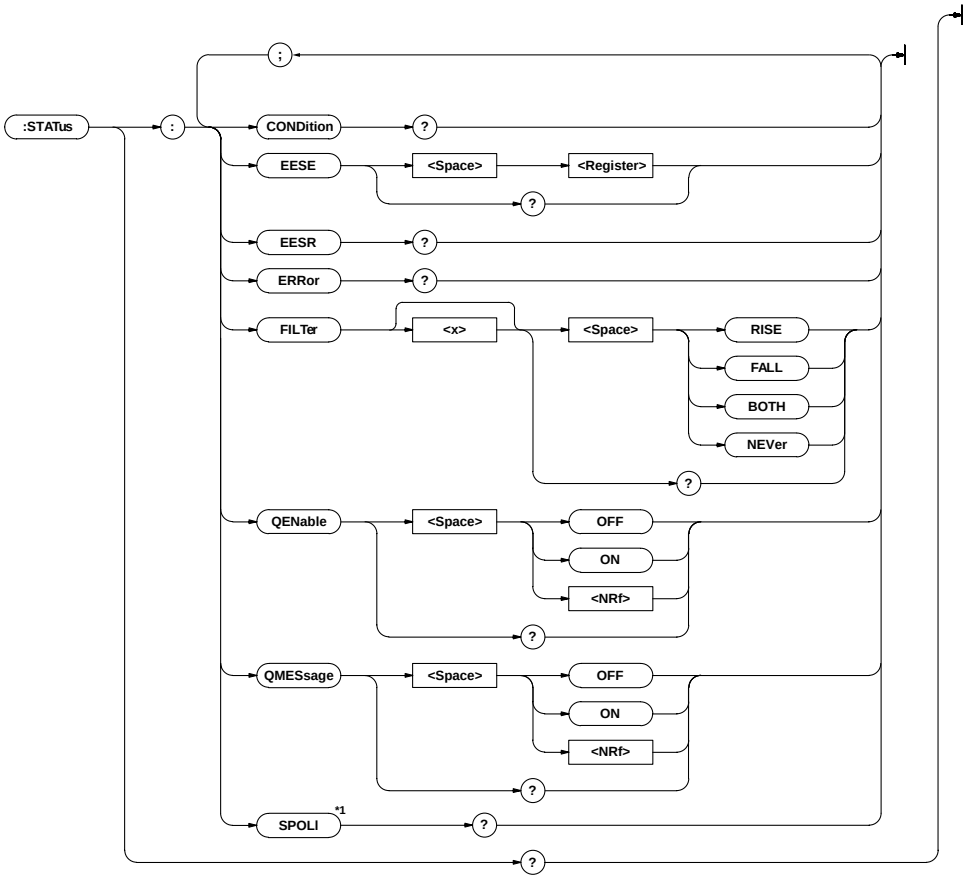


STARt

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

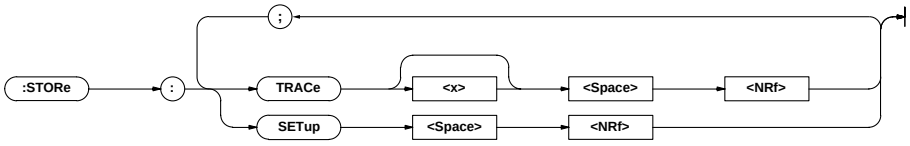
The STOP command is used to stop acquisition. The same function can be performed using the START/STOP key on the front panel.

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>

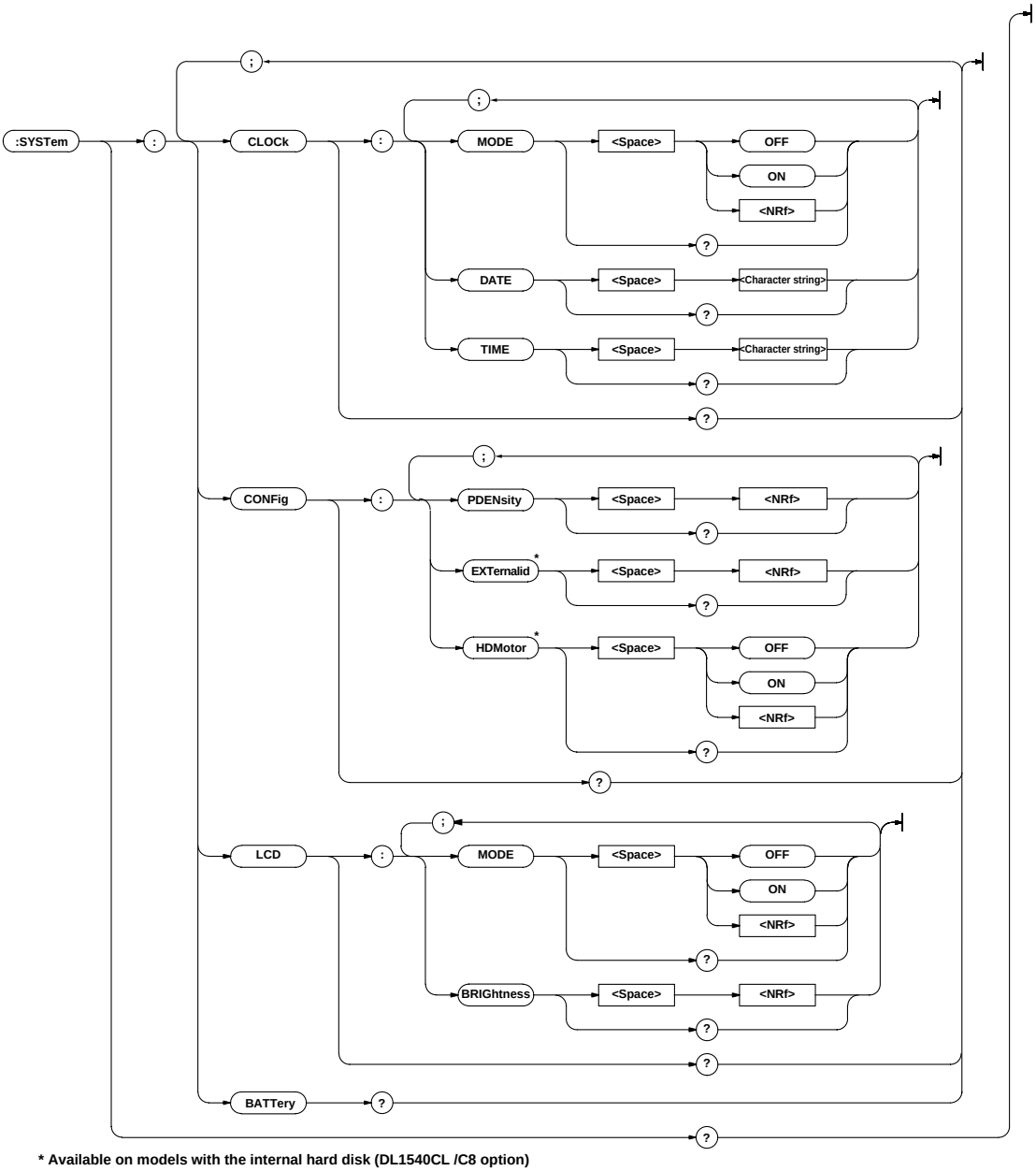
Function Stores waveform data in the internal memory.

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.



* Available on models with the internal hard disk (DL1540CL /C8 option)

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:EXTernalid

Function Sets the SCSI-ID number of the external SCSI device, or queries the current setting.
Syntax SYSTem:CONFig:EXTernalid {<NRf>}
SYSTem:CONFig:EXTernalid?
<NRf> = 0 to 3,5
Example SYSTEM:CONFIG:EXTERNALID 1
SYSTEM:CONFIG:EXTERNALID?→:SYSTEM:
CONFIG:EXTERNALID 1
Description The SCSI-ID number cannot be changed, if the internal hard disk is not present.
This command can only be used on models with the internal hard disk.

SYSTem:CONFig:HDMotor

Function Turns ON/OFF the internal hard disk (option) motor, or queries the current setting.
Syntax SYSTem:CONFig:HDMotor {<Boolean>}
SYSTem:CONFig:HDMotor?
Example SYSTEM:CONFIF:HDMOTOR ON
SYSTEM:CONFIF:HDMOTOR?→:SYSTEM:CONFIF:
HDMOTOR 1
Description An error occurs if the internal hard disk (option) is not present.

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

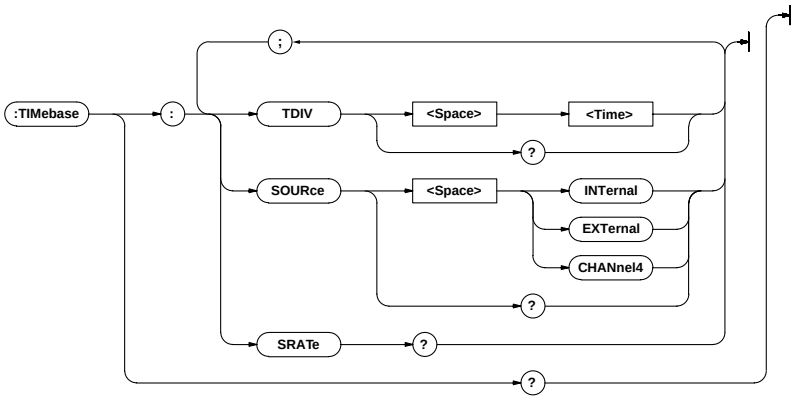
Function Sets the brightness of the LCD, or queries the setting.
Syntax SYSTem:LCD:BRIGhtness {<NRf>}
SYSTem:LCD:BRIGhtness?
<NRf>=1 to 4
Example SYSTEM:LCD:BRIGTHNESS 2
SYSTEM:LCD:BRIGTHNESS?→:SYSTEM:LCD:
BRIGTHNESS 2

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

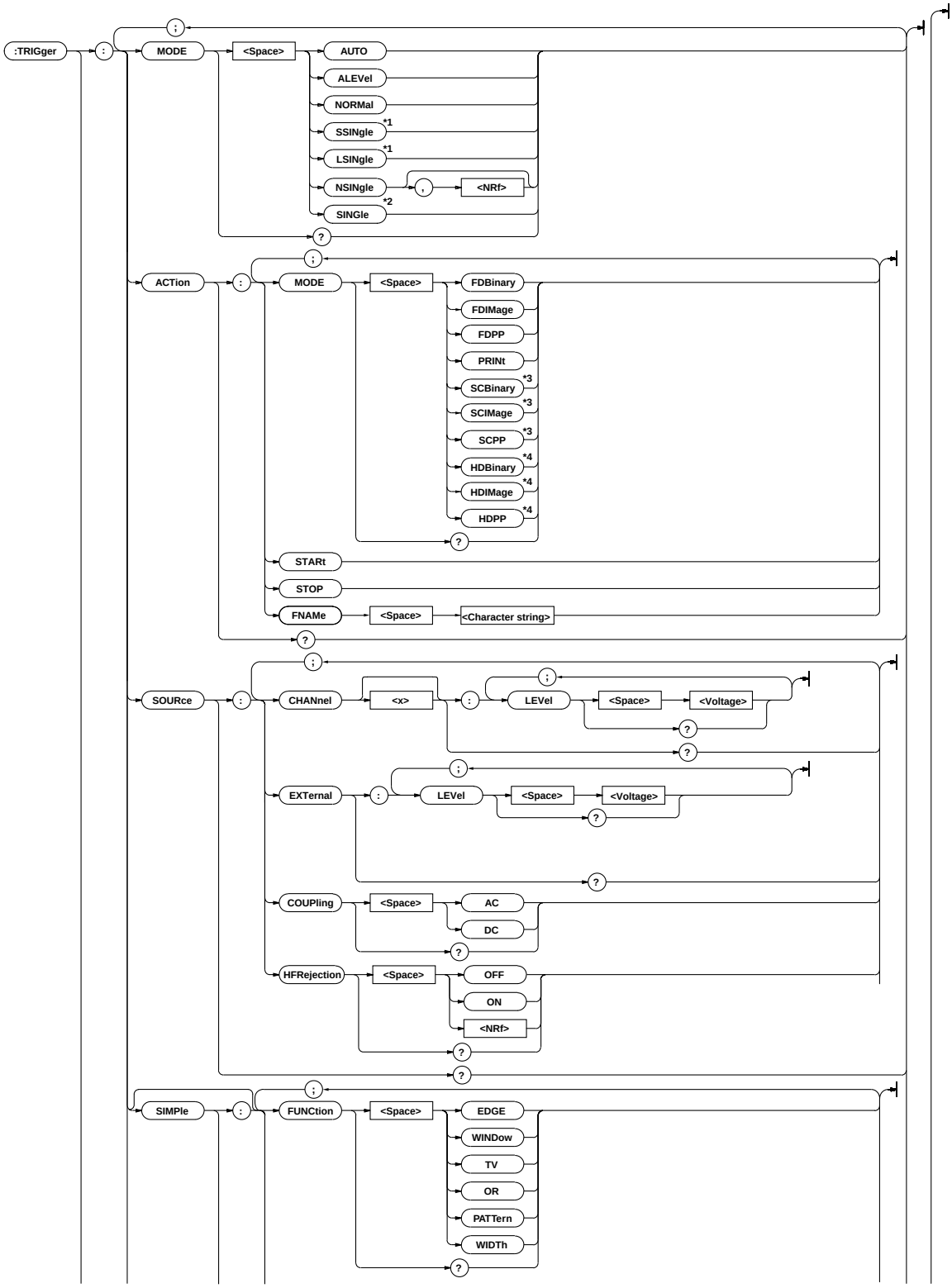
Function Set the Time axis, or queries the current setting.

Syntax TIMebase:TDIV <Time>
TIMebase:TDIV?
<Time>=2ns to 50s(in steps of 1-2-5)

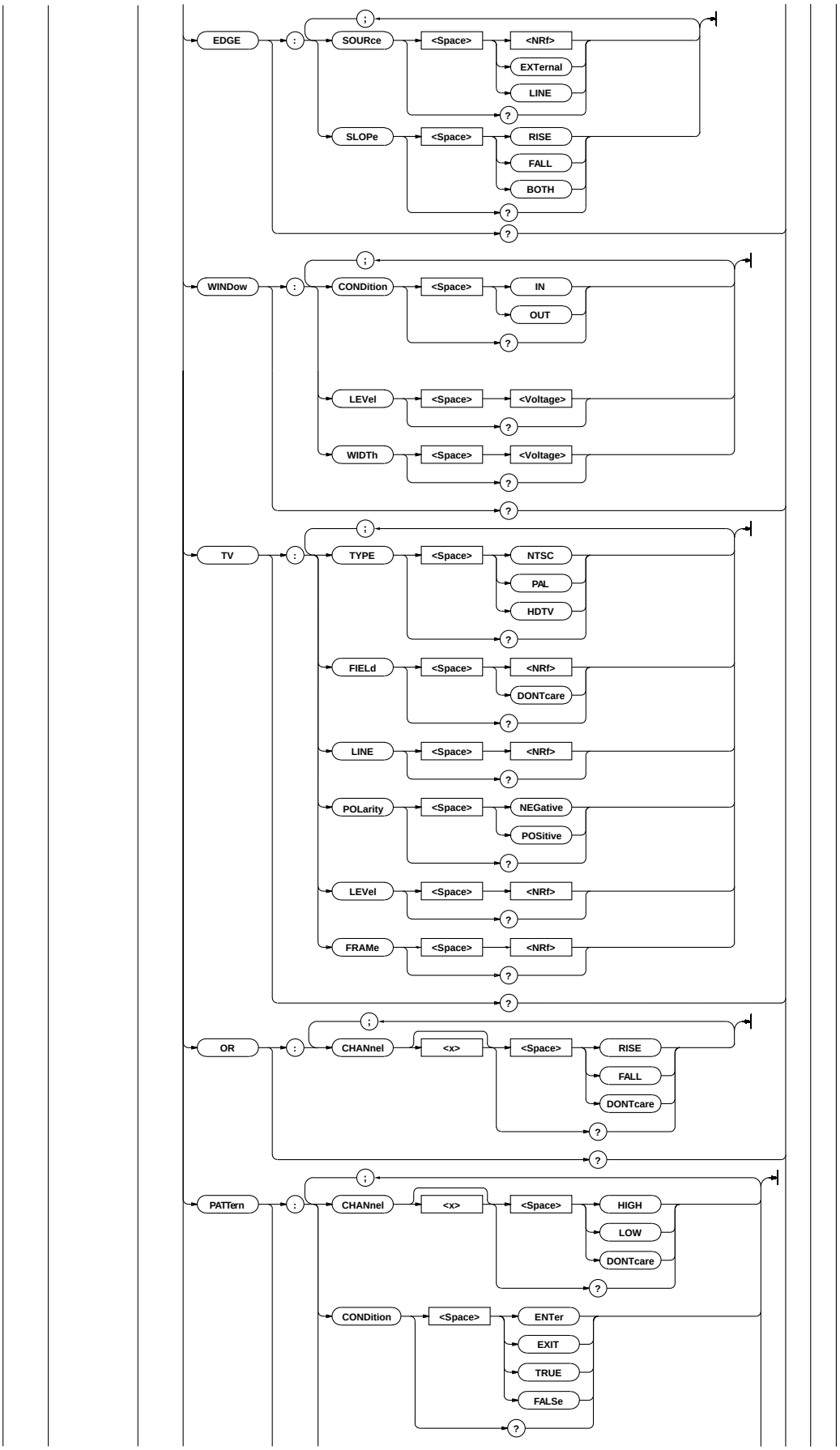
Example TIMEBASE:TDIV 1MS
TIMEBASE:TDIV?→:TIMEBASE:
TDIV 1.000000E-03

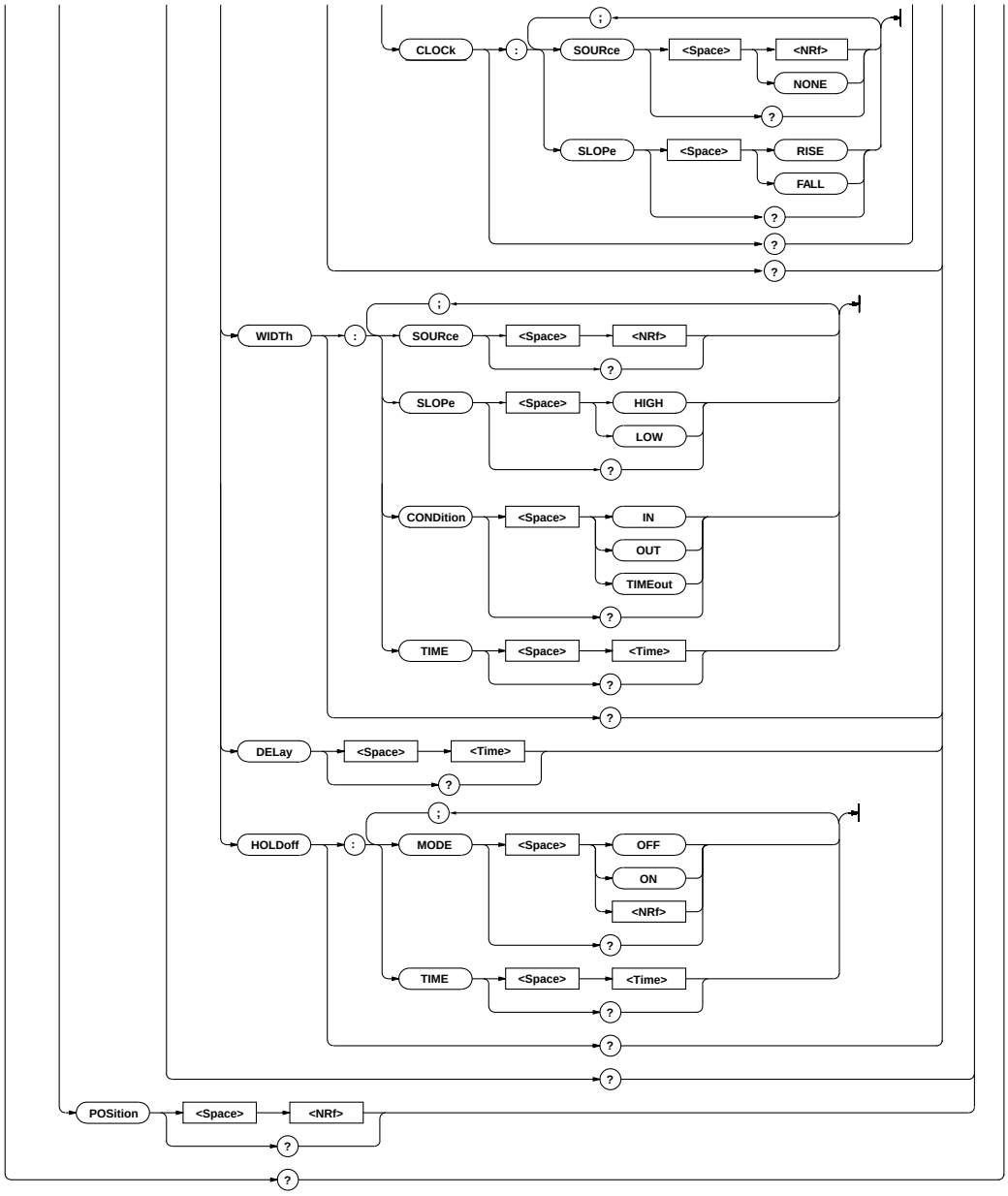
4.27 TRIGger Group

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 on models with the internal hard disk (DL1540CL /C8 option) or through the SCSI interface unit 700930 (sold separately).
*4 Available on models with the internal hard disk (DL1540CL /C8 option)


EDGE
:
;
SOURce
<Space>
<NRf>
EXTernal
LINE
?
SLOPe
<Space>
RISE
FALL
BOTH
?
?
WINDow
:
;
CONDition
<Space>
IN
OUT
?
LEVel
<Space>
<Voltage>
?
WIDTh
<Space>
<Voltage>
?
?
TV
:
;
TYPE
<Space>
NTSC
PAL
HDTV
?
FIELd
<Space>
<NRf>
DONTcare
?
LINE
<Space>
<NRf>
?
POLarity
<Space>
NEGative
POSitive
?
LEVel
<Space>
<NRf>
?
FRAMe
<Space>
<NRf>
?
?
OR
:
;
CHANnel
<x>
<Space>
RISE
FALL
DONTcare
?
?
PATTern
:
;
CHANnel
<x>
<Space>
HIGH
LOW
DONTcare
?
CONDition
<Space>
ENTer
EXIT
TRUE
FALSe
?



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|HDPP|
HDBinary|HDIMage}
TRIGger:ACTion:MODE?
```

Example
```
TRIGGER:ACTION:MODE FDPP
TRIGGER:ACTION:MODE?→:TRIGGER:ACTION:
MODE FDPP
```

Description
- SCPP, SCBinary, and SCIMage can be used through the SCSI interface unit 700930 (sold separately) or the SCSI interface connector (/C8 option).
- SCPP, SCBinary and SCIMage are available only when the SCSI is connected.
- HDPP, HDBinary, and HDIMage are available only when the internal hard disk (option) is present.

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

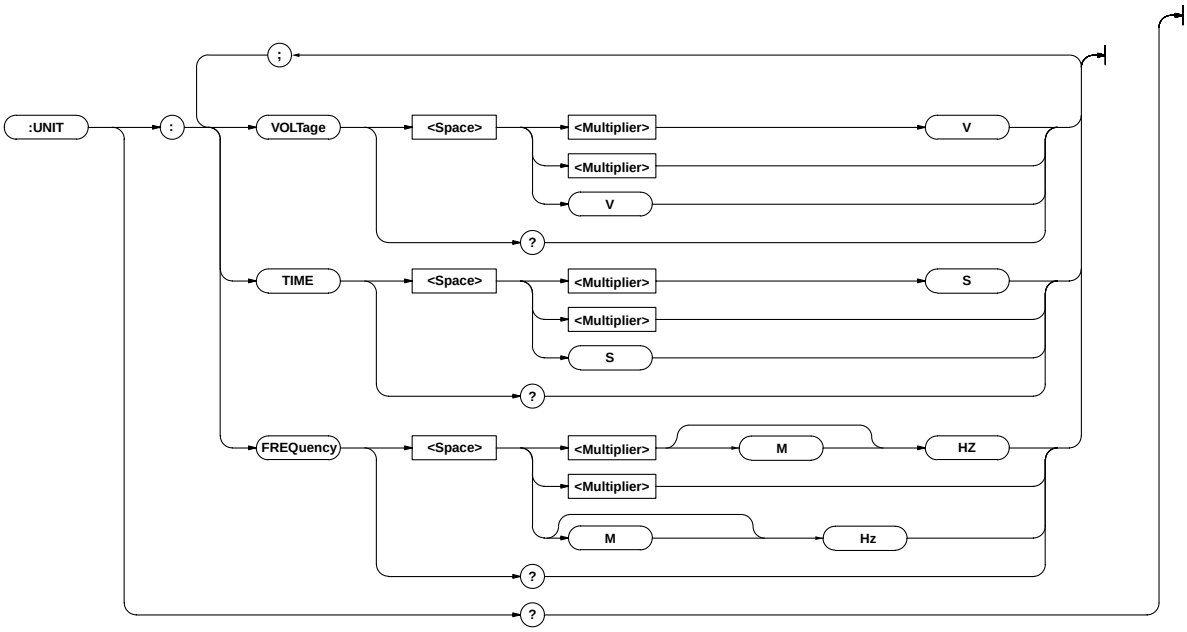
Function Turns the HF rejection ON/OFF, or queries the current setting.

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

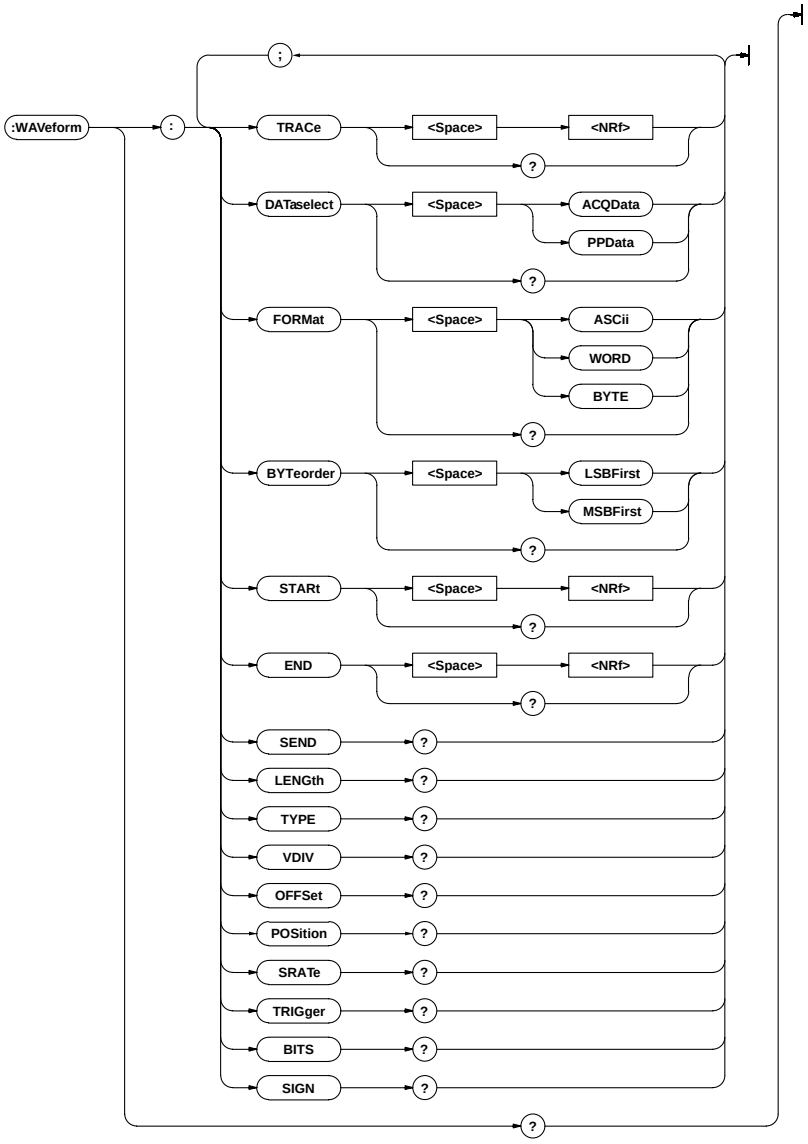
Function Sets the default value for <Voltage>, or queries the current setting.

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:
  $$\text{Voltage} = \frac{\text{V/div x Byte data}}{12.5} + \text{Offset}$$
  "WAVeform:BITS?" will return "9".
  (2) If data is unsigned:
  $$\text{Voltage} = \frac{\text{V/div x \{Byte data-Position\}}}{25} + \text{Offset}$$
  "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.
  $$\text{Voltage} = \frac{\text{V/div x Word data}}{3200} + \text{Offset}$$
  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 as 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?

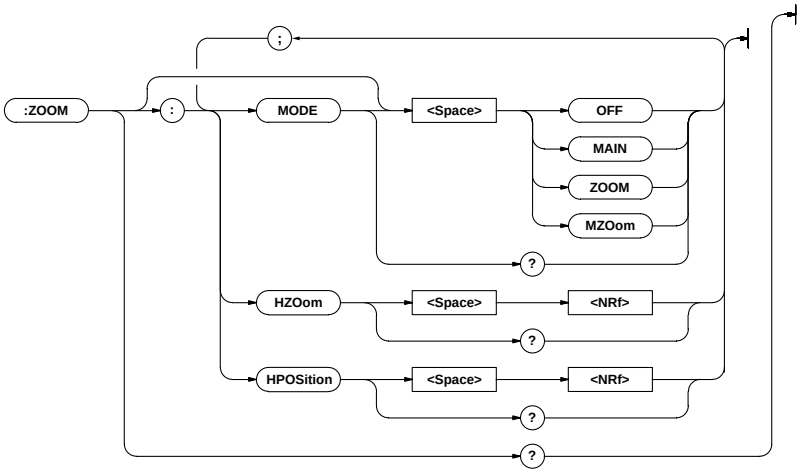
Function Queries the V/div value for the waveform designated by "WAVeform:TRACe".

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]

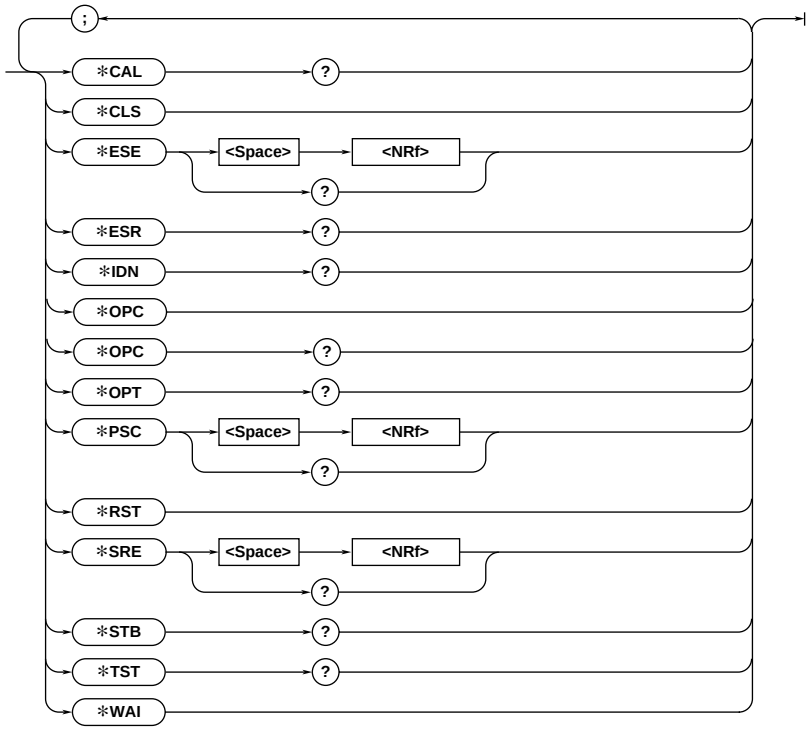
Function Sets the display type for the expanded waveform, or queries the current setting.

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 installled.
- "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 filer. 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.

*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

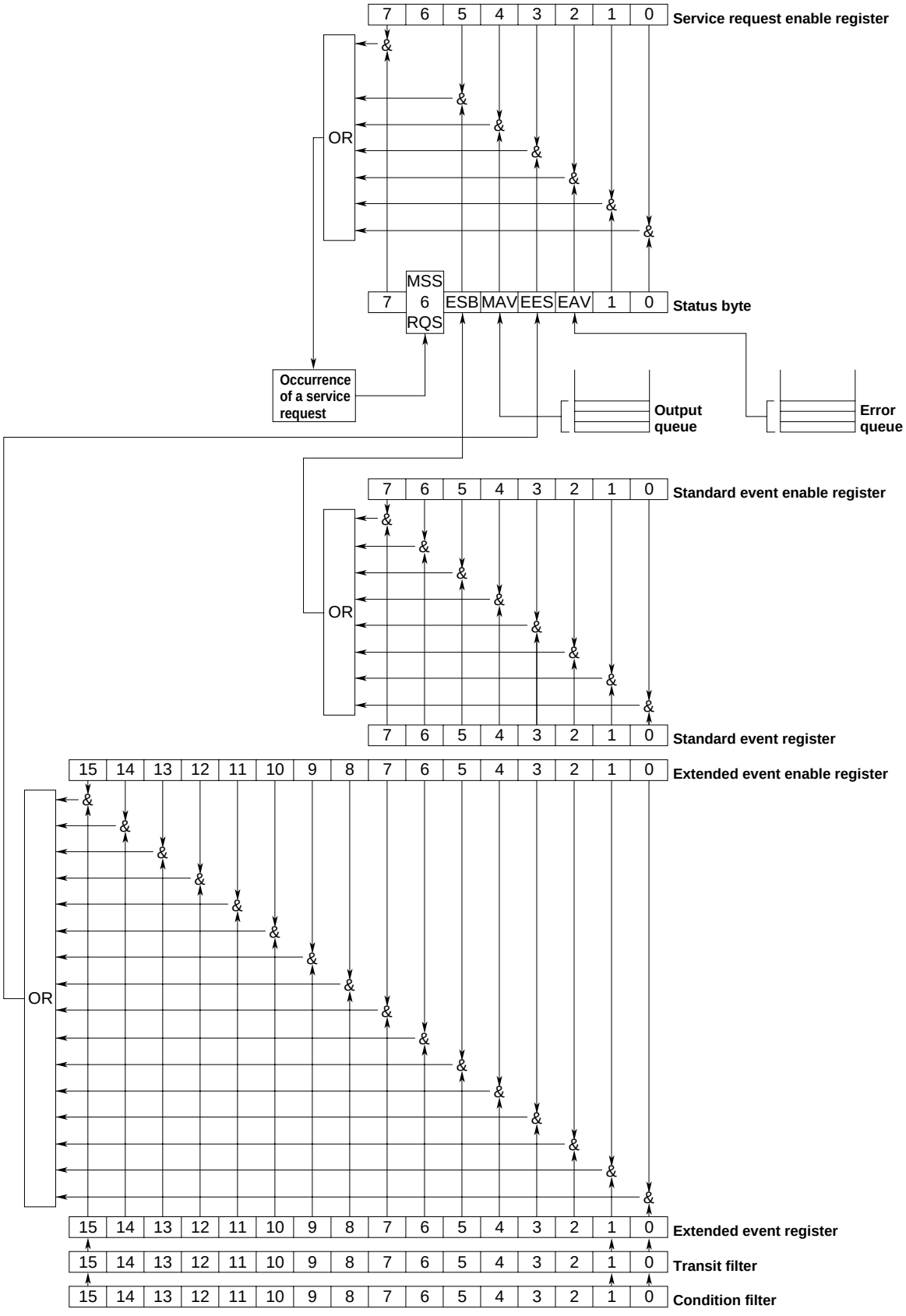
Example *WAI

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:EESR?
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

7	6 (RQS / MSS)	ESB	MAV	EES	EAV	1	0

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.

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.

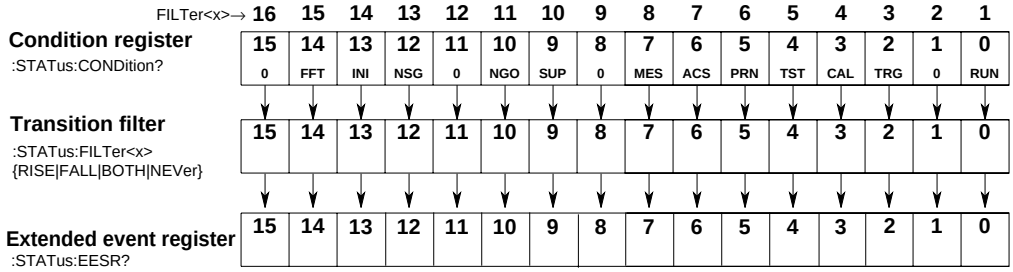
Clearing the Standard Event Register

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 seperately, 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

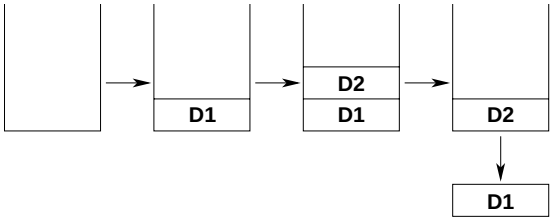
Overview of the Output Queue

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

6.1 Before Programming

Environment

Model: MS-DOS/V Computer
Language: Visual BASIC Ver5.0 Peofessional Edition or more

Setting up the Visual Basic

Compornent: MSComm
Standardmodule: Niglobal.bas
Vbib/32.bas

Setting up the Instrument

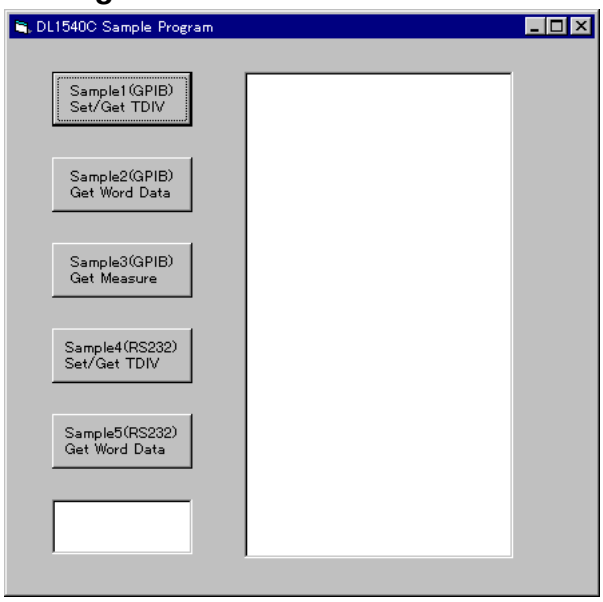
- **Initialization**
  None of the sample programs given in this chapter include initialization of the instrument, so be sure to initialize the instrument using the Initialize menu before running the programs.

- **GPIB**
  All the sample programs given in this chapter use address 1 for the instrument, so be sure to assign the instrument to address 1 as described on page 1-4.

- **RS232**
  All the sample programs in this chapter assume the below settings, so be sure parameters as described on page 2-7.

Baud Rate	19200
Handshake	CTS-RTS
Parity Check	NO
Stop Bit	1
Character Length	8
Terminator	LF

6.2 Image of Sample Progaram



6.3 Initialize/Error/Execute

```
Option Explicit
Dim StartFlag As Integer                        'Start Flag
Dim TimerCount As Integer                       'Timeout(RS232)
Dim Addr As Integer                             'GPIB Address
Dim Timeout As Integer                          'Timeout
Dim Dev As Integer                              'Device ID(GPIB)
Dim CtsFlag As Integer                          'CTS Flag
Dim Term As String                              'Terminator
Dim Query(3) As String                          'Query String
Dim WaveBuffer(20070) As Integer                'WaveData Buffer(GPIB)
Dim Dummy As Integer
```

```
Private Function InitGpib() As Integer
    Dim eos As Integer                          'EOS
    Dim eot As Integer                          'EOI
    Dim brd As Integer                          'GPIB Board ID
    Dim sts As Integer

    eos = &HC0A                                 'Terminator = LF
    eot = 1                                     'EOI = Enable
    Term = Chr(10)
    Timeout = T10s                              'Timeout = 10s

    brd = ilfind("GPIB0")
    If (brd < 0) Then
        Call DisplayGPIBError(brd, "ilfind")
        InitGpib = 1
        Exit Function
    End If
    Dev = ildev(0, Addr, 0, Timeout, eot, eos)
    If (Dev < 0) Then
        Call DisplayGPIBError(Dev, "ildev")
        InitGpib = 1
        Exit Function
    End If
    sts = ilsic(brd)                            'Set IFC
    If (sts < 0) Then
        Call DisplayGPIBError(sts, "ilsic")
        InitGpib = 1
        Exit Function
    End If
    InitGpib = 0
End Function
```

```
Private Function InitSerial() As Integer
    Dim rat As String

    MSComm1.CommPort = 1                        'Port = COM1

    rat = "9600,N,8,1"                          'Rate = 9600, NoParity, 8Bit,
1Stopbit

    MSComm1.Settings = rat

    MSComm1.Handshaking = comRTS                'Handshake = CTS-RTS
    MSComm1.RTSEnable = True                    'RTS = TRUE
    CtsFlag = 1
    Term = Chr(10)                              'Terminator = LF
    Timeout = 10                                'Timeout = 10s
    InitSerial = 0
End Function
```

```
Private Sub DisplayGPIBError(ByVal sts As Integer, ByVal msg As String)
    Dim wrn As String
    Dim ers As String
    Dim ern As Integer

    If (sts And TIMO) Then
        wrn = "Time out" + Chr(13)
    Else
        wrn = ""
    End If
    If (sts And EERR) Then
        ern = iberr
        If (ern = EDVR) Then
            ers = "EDVR:System error"
        ElseIf (ern = ECIC) Then
            ers = "ECIC:Function requires GPIB board to be CIC"
        ElseIf (ern = ENOL) Then
            ers = "ENOL:No Listeners on the GPIB"
        ElseIf (ern = EADR) Then
            ers = "EADR:GPIB board not addressed correctly"
        ElseIf (ern = EARG) Then
            ers = "EARG:Invalid argument to function call"
        ElseIf (ern = ESAC) Then
            ers = "ESAC:GPIB board not System Controller as required"
        ElseIf (ern = EABO) Then
            ers = "EABO:I/O operation aborted(timeout)"
        ElseIf (ern = ENEB) Then
            ers = "ENEB:Nonexistent GPIB board"
        ElseIf (ern = EDMA) Then
            ers = "EDMA:DMA error"
        ElseIf (ern = EOIP) Then
            ers = "EOIP:I/O operation started before previous operation completed"
        ElseIf (ern = ECAP) Then
            ers = "ECAP:No capability for intended operation"
        ElseIf (ern = EFSO) Then
            ers = "EFSO:File system operation error"
        ElseIf (ern = EBUS) Then
            ers = "EBUS:GPIB bus error"
        ElseIf (ern = ESTB) Then
            ers = "ESTB:Serial poll status byte queue overflow"
        ElseIf (ern = ESRQ) Then
            ers = "ESRQ:SRQ remains asserted"
        ElseIf (ern = ETAB) Then
            ers = "ETAB:The return buffer is full"
        ElseIf (ern = ELCK) Then
            ers = "ELCK:Address or board is locked"
        Else
            ers = ""
        End If
    Else
        ers = ""
    End If

    MsgBox ("Status No. " + Str(sts) + Chr(13) + wrn + "Error No. " + Str(ern) + Chr(13)
+ ers + Chr(13) + msg), vbExclamation, "Error!"
    Call ibonl(Dev, 0)
    Dev = -1
End Sub
```

```
Private Sub DIsplayRS232Error(ByVal erm As String, Optional ByVal msg As String = "")
    MsgBox (erm + Chr(13) + msg), vbExclamation, "Error!"
End Sub
```

```
Private Sub Command1_Click()
    Dim sts As Integer

    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    Dummy = DoEvents()
    sts = GpibTdiv                              'Run Sample1(GPIB) Set/Get TDIV
    If (sts = 0) Then
        List1.AddItem Query(0)
    End If
    Text1.Text = "END"
    StartFlag = 0
End Sub
```

```
Private Sub Command2_Click()
    Dim sts As Integer

    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    Dummy = DoEvents()
    sts = GpibGetWord                           'Run Sample2(GPIB) Get Word Data
    If (sts = 0) Then
        List1.AddItem "END"
    End If
    Text1.Text = "END"
    StartFlag = 0
End Sub
```

```
Private Sub Command3_Click()
    Dim sts As Integer

    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    Dummy = DoEvents()
    sts = GpibGetMeasure                        'Run Sample3(GPIB) Get Measure
    If (sts = 0) Then
        List1.AddItem Query(0)
        List1.AddItem Query(1)
        List1.AddItem Query(2)
    End If
    Text1.Text = "END"
    StartFlag = 0
End Sub
```

```
Private Sub Command4_Click()
    Dim sts As Integer

    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    sts = RS232Tdiv                             'Run Sample4(RS232) Set/Get TDIV
    If (sts = 0) Then
        List1.AddItem Query(0)
    End If
```

```
    Text1.Text = "END"
    StartFlag = 0
End Sub
```

```
Private Sub Command5_Click()
    Dim sts As Integer

    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    sts = RS232GetWord                          'Run Sample5(RS232) Get Word Data
    If (sts = 0) Then
        List1.AddItem "END"
    End If
    Text1.Text = "END"
    StartFlag = 0
End Sub
```

```
Private Sub Form_Load()

    StartFlag = 0                               'Clear Start Flag
    Dev = -1                                    'Clear device id
    Addr = 1                                    'GPIB Address = 1
    Timer1.Interval = 0
    Command1.Caption = "Sample1(GPIB)" + Chr(13) + "Set/Get TDIV"
    Command2.Caption = "Sample2(GPIB)" + Chr(13) + "Get Word Data"
    Command3.Caption = "Sample3(GPIB)" + Chr(13) + "Get Measure"
    Command4.Caption = "Sample4(RS232)" + Chr(13) + "Set/Get TDIV"
    Command5.Caption = "Sample5(RS232)" + Chr(13) + "Get Word Data"
    Text1.Text = ""

End Sub
```

```
Private Sub MSComm1_OnComm()
    Dim evt As Integer

    evt = MSComm1.CommEvent
    Select Case evt
    'Error
        Case comBreak
            Call DIsplayRS232Error("comBreak:Break received")
        Case comCDTO
            Call DIsplayRS232Error("comCDTO CD(RLSD) timeout")
        Case comCTSTO
            Call DIsplayRS232Error("comCTSTO:CTS timeout")
        Case comDSRTO
            Call DIsplayRS232Error("commDSRTO:DSR timeout")
        Case comFrame
            Call DIsplayRS232Error("comFrame:Frame error")
        Case comOverrun
            Call DIsplayRS232Error("comOverrun:Overrun")
        Case comRxOver
            Call DIsplayRS232Error("comRxOver:Receive buffer overflow")
        Case comRxParity
            Call DIsplayRS232Error("commRxParity:Parity error")
        Case comTxFull
            Call DIsplayRS232Error("comTxFull:Send buffer overflow")
    'Event
        Case comEvReceive
        Case comEvCD
        Case comEvCTS
        Case comEvDSR
        Case comEvRing
        Case comEvSend
    End Select
```

```
End Sub

Private Sub Timer1_Timer()
    TimerCount = TimerCount + 1
End Sub
```

6.4 Sets/Queries the T/Div

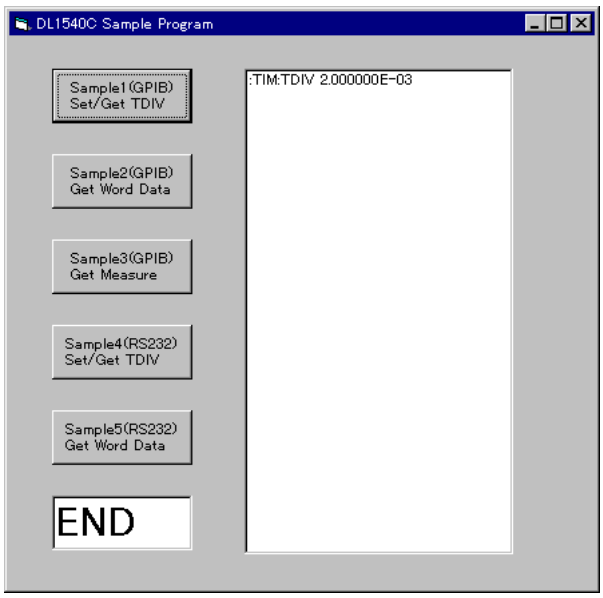
Sample1(GPIB) Set/Get TDIV

```
Private Function GpibTdiv() As Integer
    Dim msg As String                               'Command buffer
    Dim qry As String                               'Query biffer
    Dim sts As Integer

    msg = Space$(100)
    qry = Space$(100)

    sts = InitGpib                                  'Initialize GPIB
    If (sts <> 0) Then
        GpibTdiv = 1
        Exit Function
    End If

    msg = "TIMEBASE:TDIV 2ms" + Term                'Set T/div = 2ms
    sts = ilwrt(Dev, msg, Len(msg))                 'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibTdiv = 1
        Exit Function
    End If
    msg = "TIMEBASE:TDIV?" + Term                   'Get T/div value
    sts = ilwrt(Dev, msg, Len(msg))                 'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibTdiv = 1
        Exit Function
    End If
    sts = ilrd(Dev, qry, Len(qry))                  'Receive Query
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibTdiv = 1
        Exit Function
    End If
    Query(0) = Left(qry, ibcntl - 1)
    Call ibonl(Dev, 0)
    GpibTdiv = 0
End Function
```

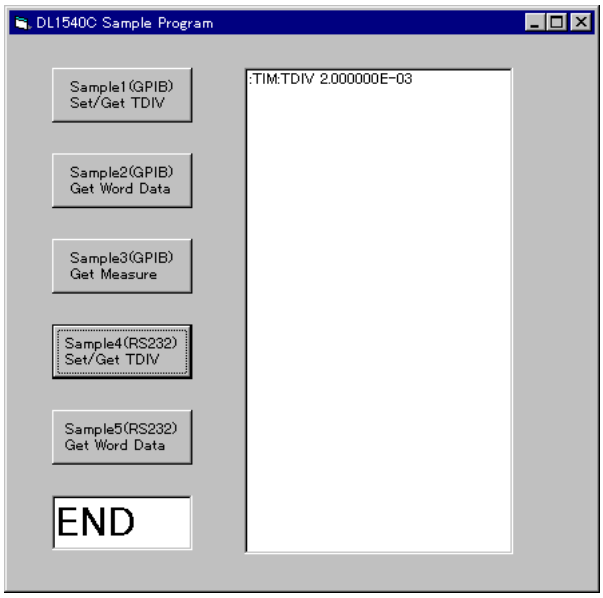
```
Sample4(RS232) Set/Get TDIV
Private Function RS232Tdiv() As Integer
    Dim msg As String                               'Command buffer
    Dim qry As String                               'Query biffer
    Dim sts As Integer

    msg = Space$(100)
    qry = CStr(Empty)

    sts = InitSerial                                'Initialize RS232
    If (sts <> 0) Then
        Exit Function
    End If

    MSComm1.InputLen = 0                            'Receive All Data
    MSComm1.InputMode = comInputModeText            'Text Mode
    MSComm1.PortOpen = True                         'Port Open
    MSComm1.OutBufferCount = 0                      'Out Buffer Clear
    MSComm1.InBufferCount = 0                       'In Buffer Clear
    Timer1.Interval = 1000

    If CtsFlag = 1 Then                             'If CTS = FALSE
        TimerCount = 1                              'Wait until CTS = TRUE
        Do
            Dummy = DoEvents()
            If (TimerCount >= Timeout) Then
                Call DIsplayRS232Error("CTS Timeout")
```

```
            RS232Tdiv = 1
            GoTo finish
        End If
    Loop Until MSComm1.CTSHolding = True
End If

msg = "TIMEBASE:TDIV 2ms" + Term                'Set T/div = 2ms
MSComm1.Output = msg                            'Send Command

TimerCount = 1
Do                                              'Wait until OutBufferCount = 0
    Dummy = DoEvents()
    If (TimerCount >= Timeout) Then
        Call DIsplayRS232Error("Send Timeout", msg)
        RS232Tdiv = 1
        GoTo finish
    End If
Loop Until MSComm1.OutBufferCount = 0

msg = "TIMEBASE:TDIV?" + Term                   'Get T/div value
MSComm1.Output = msg                            'Send Command

TimerCount = 1
Do                                              'Wait until OutBufferCount = 0
    Dummy = DoEvents()
    If (TimerCount >= Timeout) Then
        Call DIsplayRS232Error("Send Timeout", msg)
        RS232Tdiv = 1
        GoTo finish
    End If
Loop Until MSComm1.OutBufferCount = 0

TimerCount = 1
Do                                              'Receive Query
    qry = qry + MSComm1.Input                   'Wait until End Data = Terminator
    Dummy = DoEvents()
    If (TimerCount >= Timeout) Then
        Call DIsplayRS232Error("Receive Timeout", msg)
        RS232Tdiv = 0
        GoTo finish
    End If
Loop Until Right$(qry, 1) = Term

Query(0) = Left$(qry, Len(qry) - 1)
RS232Tdiv = 0
finish:
MSComm1.PortOpen = False                        'Port Close
Timer1.Interval = 0

End Function
```

GPIB


DL1540C Sample Program
Sample1(GPIB)
Set/Get TDIV
Sample2(GPIB)
Get Word Data
Sample3(GPIB)
Get Measure
Sample4(RS232)
Set/Get TDIV
Sample5(RS232)
Get Word Data
END
:TIM:TDIV 2.000000E-03

RS232


DL1540C Sample Program
Sample1(GPIB)
Set/Get TDIV
Sample2(GPIB)
Get Word Data
Sample3(GPIB)
Get Measure
Sample4(RS232)
Set/Get TDIV
Sample5(RS232)
Get Word Data
END
:TIM:TDIV 2.000000E-03

6.5 Data Output in Word Format

Sample2(GPIB) Get Word Data

```
Private Function GpibGetWord() As Integer
    Dim msg As String                                'Command buffer
    Dim qry As String                                'Query biffer
    Dim sts As Integer
    Dim vdv As Variant                               'Vdiv value
    Dim eos As Integer                               'EOS
    Dim hlg As Integer                               'Block Header Length
    Dim dlg As Integer                               'Blocl Data Length
    Dim dat As Variant                               'Data
    Dim i   As Integer

    msg = Space$(100)
    qry = Space$(100)

    sts = InitGpib                                   'Initialize GPIB
    If (sts <> 0) Then
        GpibGetWord = 1
        Exit Function
    End If

    msg = "STOP" + Term                             'Stop Acquisition
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetWord = 1
        Exit Function
    End If
    msg = "COMMUNICATE:HEADER OFF" + Term          'Query Header Off(for Get V/div)
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetWord = 1
        Exit Function
    End If
    msg = "WAVEFORM:TRACE 1" + Term                'Trace = 1
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetWord = 1
        Exit Function
    End If
    msg = "WAVEFORM:DATASELECT ACQDATA" + Term     'Data = Acquisition Data
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetWord = 1
        Exit Function
    End If
    msg = "WAVEFORM:FORMAT WORD" + Term             'Data Format = WORD
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetWord = 1
        Exit Function
    End If
    msg = "WAVEFORM:BYTEORDER LSBFIRST" + Term     'Data Byte order = LSB First(for
Little Endian)
 sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetWord = 1
        Exit Function
    End If
    msg = "WAVEFORM:START 0;END 10031" + Term      'START 0,END 10031(Length = 10032)
    sts = ilwrt(Dev, msg, Len(msg))
```

```
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetWord = 1
    Exit Function
End If
msg = "WAVEFORM:VDIV?" + Term                   'Get V/div value
sts = ilwrt(Dev, msg, Len(msg))
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetWord = 1
    Exit Function
End If
sts = ilrd(Dev, qry, Len(qry))
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetWord = 1
    Exit Function
End If
vdv = Val(qry)

eos = 0
sts = ileos(Dev, eos)                           'Terminator = None(for Binary Data)
If (sts < 0) Then
    Call DisplayGPIBError(sts, "ileos")
    GpibGetWord = 1
    Exit Function
End If
msg = "WAVEFORM:SEND?" + Term                   'Receive Waveform Data
sts = ilwrt(Dev, msg, Len(msg))
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetWord = 1
    Exit Function
End If
sts = ilrd(Dev, qry, 2)                         'Receive "#X"
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetWord = 1
    Exit Function
End If
hlg = Val(Mid$(qry, 2, 1))
sts = ilrd(Dev, qry, hlg)                       'Receive Block Header
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetWord = 1
    Exit Function
End If
dlg = Val(Left$(qry, hlg))                      'dlg = Data Byte Length

sts = ilrdi(Dev, WaveBuffer(), dlg + 1)         'Receive Waveform Data + LF
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetWord = 1
    Exit Function
End If

For i = 0 To (dlg / 2 - 1) Step 1
    dat = WaveBuffer(i) * vdv / 3200
    List1.AddItem CStr(i) + ":" + CStr(dat)
Next i

eos = &HC0A
sts = ileos(Dev, eos)                           'Terminator = LF
If (sts < 0) Then
    Call DisplayGPIBError(sts, "ileos")
    GpibGetWord = 1
    Exit Function
End If
msg = "COMMUNICATE:HEADER ON" + Term            'Query Header On
```

```
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetWord = 1
        Exit Function
    End If
    Call ibonl(Dev, 0)
    GpibGetWord = 0
End Function
```

Sample5(RS232) Get Word Data

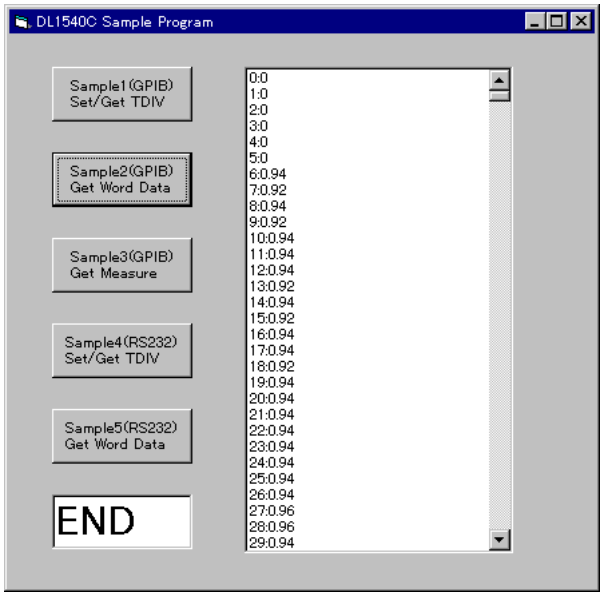
```
Private Function RS232GetWord() As Integer
    Dim msg As String                               'Command buffer
    Dim qry As String                               'Query biffer
    Dim sts As Integer
    Dim vdv As Variant                              'V/div value
    Dim hlg As Integer                              'Block Header Length
    Dim dlg As Integer                              'Blocl Data Length
    Dim buf As Variant                              'temporary buffer
    Dim dat As Variant                              'data buffer
    Dim i   As Integer

    msg = Space$(100)
    qry = CStr(Empty)

    sts = InitSerial                                'Initialize RS232
    If (sts <> 0) Then
        Exit Function
    End If

    MSComm1.InputLen = 0                            'Receive All Data
    MSComm1.InputMode = comInputModeText            'Text Mode
    MSComm1.PortOpen = True                         'Port Open
    MSComm1.OutBufferCount = 0                      'Out Buffer Clear
    MSComm1.InBufferCount = 0                       'In Buffer Clear
    Timer1.Interval = 1000

    If CtsFlag = 1 Then                             'If CTS = FALSE
        TimerCount = 1                              'Wait unitl CTS = TRUE
        Do
            Dummy = DoEvents()
            If (TimerCount >= Timeout) Then
                Call DIsplayRS232Error("CTS Timeout")
                RS232GetWord = 1
                GoTo finish
            End If
        Loop Until MSComm1.CTSHolding = True
    End If

    msg = "STOP" + Term                             'Stop Acquisition
    MSComm1.Output = msg
    TimerCount = 1
    Do
        Dummy = DoEvents()
        If (TimerCount >= Timeout) Then
            Call DIsplayRS232Error("Send Timeout", msg)
            RS232GetWord = 1
            GoTo finish
        End If
    Loop Until MSComm1.OutBufferCount = 0

    msg = "COMMUNICATE:HEADER OFF" + Term           'Query Header Off(for Get V/div)
    MSComm1.Output = msg
    TimerCount = 1
    Do
        Dummy = DoEvents()
        If (TimerCount >= Timeout) Then
            Call DIsplayRS232Error("Send Timeout", msg)
            RS232GetWord = 1
            GoTo finish
        End If
    Loop Until MSComm1.OutBufferCount = 0

    msg = "WAVEFORM:TRACE 1" + Term                 'Trace = 1
    MSComm1.Output = msg
    TimerCount = 1
```

```
    Do
        Dummy = DoEvents()
        If (TimerCount >= Timeout) Then
            Call DIsplayRS232Error("Send Timeout", msg)
            RS232GetWord = 1
            GoTo finish
        End If
    Loop Until MSComm1.OutBufferCount = 0

    msg = "WAVEFORM:DATASELECT ACQDATA" + Term     'Data = Acquisition Data
    MSComm1.Output = msg
    TimerCount = 1
    Do
        Dummy = DoEvents()
        If (TimerCount >= Timeout) Then
            Call DIsplayRS232Error("Send Timeout", msg)
            RS232GetWord = 1
            GoTo finish
        End If
    Loop Until MSComm1.OutBufferCount = 0

    msg = "WAVEFORM:FORMAT WORD" + Term            'Data Format = WORD
    MSComm1.Output = msg
    TimerCount = 1
    Do
        Dummy = DoEvents()
        If (TimerCount >= Timeout) Then
            Call DIsplayRS232Error("Send Timeout", msg)
            RS232GetWord = 1
            GoTo finish
        End If
    Loop Until MSComm1.OutBufferCount = 0

    msg = "WAVEFORM:BYTEORDER LSBFIRST" + Term     'Data Byte order = LSB First(for
Little Endian)
    MSComm1.Output = msg
    TimerCount = 1
    Do
        Dummy = DoEvents()
        If (TimerCount >= Timeout) Then
            Call DIsplayRS232Error("Send Timeout", msg)
            RS232GetWord = 1
            GoTo finish
        End If
    Loop Until MSComm1.OutBufferCount = 0

    msg = "WAVEFORM:START 0;END 10031" + Term      'START 0,END 10031(Length = 10032)
    MSComm1.Output = msg
    TimerCount = 1
    Do
        Dummy = DoEvents()
        If (TimerCount >= Timeout) Then
            Call DIsplayRS232Error("Send Timeout", msg)
            RS232GetWord = 1
            GoTo finish
        End If
    Loop Until MSComm1.OutBufferCount = 0

    msg = "WAVEFORM:VDIV?" + Term                  'Get V/div value
    MSComm1.Output = msg
    TimerCount = 1
    Do
        Dummy = DoEvents()
        If (TimerCount >= Timeout) Then
            Call DIsplayRS232Error("Send Timeout", msg)
            RS232GetWord = 1
            GoTo finish
        End If
    Loop Until MSComm1.OutBufferCount = 0
    TimerCount = 1
```

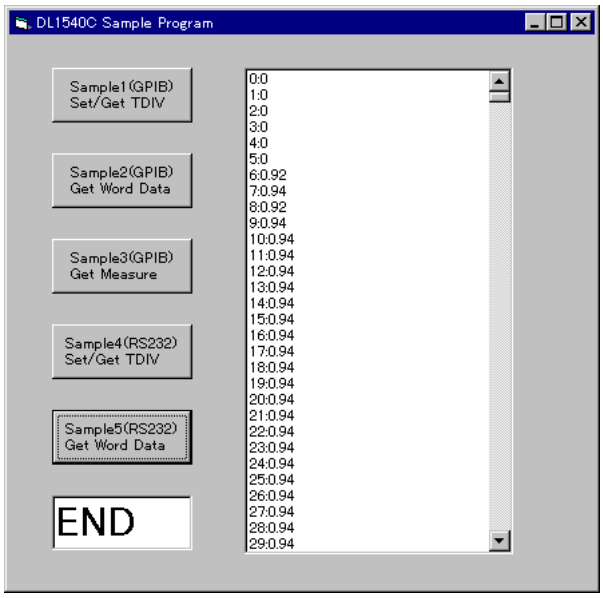
```
Do
    qry = qry + MSComm1.Input
    Dummy = DoEvents()
    If (TimerCount >= Timeout) Then
        Call DIsplayRS232Error("Receive Timeout", msg)
        RS232GetWord = 1
        GoTo finish
    End If
Loop Until Right$(qry, 1) = Term
vdv = Val(qry)

msg = "WAVEFORM:SEND?" + Term                   'Receive Waveform Data
MSComm1.Output = msg

TimerCount = 1
Do
    Dummy = DoEvents()
    If (TimerCount >= Timeout) Then
        Call DIsplayRS232Error("Send Timeout", msg)
        RS232GetWord = 1
        GoTo finish
    End If
Loop Until MSComm1.OutBufferCount = 0

MSComm1.InputLen = 2                            'Receive "#X"
TimerCount = 1
Do Until MSComm1.InBufferCount >= 1
    Dummy = DoEvents()
    If (TimerCount >= Timeout) Then
        Call DIsplayRS232Error("Receive Timeout", msg)
        RS232GetWord = 1
        GoTo finish
    End If
Loop
qry = MSComm1.Input
hlg = Val(Mid$(qry, 2, 1))

MSComm1.InputLen = hlg                          'Receive Block Header
TimerCount = 1
Do Until MSComm1.InBufferCount >= hlg
    Dummy = DoEvents()
    If (TimerCount >= Timeout) Then
        Call DIsplayRS232Error("Receive Timeout", msg)
        RS232GetWord = 1
        GoTo finish
    End If
Loop
qry = MSComm1.Input
dlg = Val(Left$(qry, hlg))                      'leng% = Data Byte Length

MSComm1.InputMode = comInputModeBinary
MSComm1.InputLen = 2                            'Receive Waveform Data(2 Byte)

For i = 0 To (dlg / 2 - 1) Step 1               'Loop(dlg)
    TimerCount = 1
    Do Until MSComm1.InBufferCount >= 2
        Dummy = DoEvents()
        If (TimerCount >= Timeout) Then
            Call DIsplayRS232Error("Receive Timeout", msg)
            RS232GetWord = 1
            GoTo finish
        End If
    Loop
    buf = MSComm1.Input                         'Receive 1 Data(2 Byte)
    dat = buf(1) * 256 + buf(0)
    If (dat > 32767) Then
        dat = dat - 65536
    End If
    dat = dat * vdv / 3200
```

```
        List1.AddItem CStr(i) + ":" + CStr(dat)
    Next i

    msg = "COMMUNICATE:HEADER ON" + Term           'Query Header On
    MSComm1.Output = msg
    TimerCount = 1
    Do
        Dummy = DoEvents()
        If (TimerCount >= Timeout) Then
            Call DIsplayRS232Error("Send Timeout", msg)
            RS232GetWord = 1
            GoTo finish
        End If
    Loop Until MSComm1.OutBufferCount = 0

    RS232GetWord = 0
finish:
    MSComm1.PortOpen = False                       'Port Close
    Timer1.Interval = 0

End Function
```

GPIB



RS232



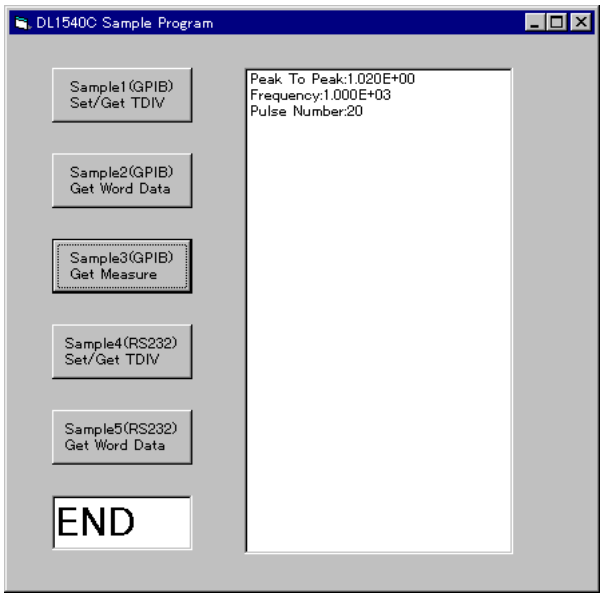
6.6 Sets/Queries Measure Value

Sample3(GPIB) GetMeasure

```
Private Function GpibGetMeasure() As Integer
    Dim msg As String                                   'Command buffer
    Dim qry As String                                   'Query biffer
    Dim sts As Integer
    Dim mdl As String

    msg = Space$(100)
    qry = Space$(100)

    sts = InitGpib                                      'Initialize GPIB
    If (sts <> 0) Then
        GpibGetMeasure = 1
        Exit Function
    End If

    msg = "COMMUNICATE:HEADER OFF" + Term           'Query Header Off(for Get V/div)
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    msg = "MEASURE:MODE OFF" + Term                 'Measure Off
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    msg = "*IDN?" + Term                            'Get DL Model
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    sts = ilrd(Dev, qry, Len(qry))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    mdl = Mid$(qry, 10, 6)
    If (StrComp(mdl, "701540") = 0) Then            'If Model = DL1540CL Then
TRIGGER:MODE SINGLE
        msg = "TRIGGER:MODE SINGLE;:STOP" + Term    'Trigger Mode = Single(S),
Acquisition = Stop
    Else
        msg = "TRIGGER:MODE SSINGLE;:STOP" + Term   'Trigger Mode = Single(S),
Acquisition = Stop
    End If
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    msg = "CHANNEL:MODE ON;INPUT:PROBE 10" + Term   'CH1 On, PROBE 10:1
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    msg = "STATUS:FILTER1 FALL" + Term              'Filter1 Fall(Start->Stop)
    sts = ilwrt(Dev, msg, Len(msg))
```

```
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetMeasure = 1
    Exit Function
End If
msg = "MEASURE:TRACE1:PTOPEAK ON;FREQUENCY ON;PNUMBER ON" + Term
                                                    'Measure P-P,Freq,Pulse On
sts = ilwrt(Dev, msg, Len(msg))
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetMeasure = 1
    Exit Function
End If
msg = "STATUS:EESR?" + Term                         'Clear Extended Event Register
sts = ilwrt(Dev, msg, Len(msg))
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetMeasure = 1
    Exit Function
End If
sts = ilrd(Dev, qry, Len(qry))
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetMeasure = 1
    Exit Function
End If

msg = "START" + Term                                'Start Acquisition(Single)
sts = ilwrt(Dev, msg, Len(msg))
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetMeasure = 1
    Exit Function
End If
msg = "COMMUNICATE:WAIT? 1" + Term                  'Wait until stop Acquisition
sts = ilwrt(Dev, msg, Len(msg))
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetMeasure = 1
    Exit Function
End If
sts = ilrd(Dev, qry, Len(qry))
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetMeasure = 1
    Exit Function
End If

msg = "MEASURE:MODE ON" + Term                      'Start Measure
sts = ilwrt(Dev, msg, Len(msg))
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetMeasure = 1
    Exit Function
End If
msg = "MEASURE:WAIT? 100" + Term                    'Wait until stop Measure
sts = ilwrt(Dev, msg, Len(msg))
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetMeasure = 1
    Exit Function
End If
sts = ilrd(Dev, qry, Len(qry))
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GpibGetMeasure = 1
    Exit Function
End If
```

```
    msg = "MEASURE:TRACE1:VALUE? PTOPEAK" + Term   'Get P-P value
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    sts = ilrd(Dev, qry, Len(qry))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    Query(0) = "Peak To Peak:" + Left$(qry, ibcntl - 1)

    msg = "MEASURE:TRACE1:VALUE? FREQUENCY" + Term 'Get Freq value
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    sts = ilrd(Dev, qry, Len(qry))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    Query(1) = "Frequency:" + Left$(qry, ibcntl - 1)

    msg = "MEASURE:TRACE1:VALUE? PNUMBER" + Term   'Get Pulse value
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    sts = ilrd(Dev, qry, Len(qry))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    Query(2) = "Pulse Number:" + Left$(qry, ibcntl - 1)
    msg = "COMMUNICATE:HEADER ON" + Term           'Query Header On
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    Call ibonl(Dev, 0)
    GpibGetMeasure = 0
End Function
```


DL1540C Sample Program
Sample1(GPIB)
Set/Get TDIV
Sample2(GPIB)
Get Word Data
Sample3(GPIB)
Get Measure
Sample4(RS232)
Set/Get TDIV
Sample5(RS232)
Get Word Data
END
Peak To Peak:1.020E+00
Frequency:1.000E+03
Pulse Number:20

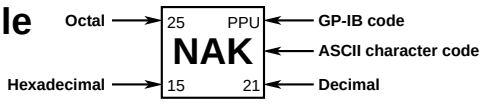
Appendix

Appendix 1 ASCII Character Codes

ASCII character codes are given below.

	0	1	2	3	4	5	6	7
0	0 **NUL** 0 0	20 **DEL** 10 16	40 0 **SP** 20 32	60 16 **0** 30 48	100 0 **@** 40 64	120 16 **P** 50 80	140 0 **‘** 60 96	160 16 **p** 70 112
1	1 GTL **SOH** 1 1	21 LLO **DC1** 11 17	41 1 **!** 21 33	61 17 **1** 31 49	101 1 **A** 41 65	121 17 **Q** 51 81	141 1 **a** 61 97	161 17 **q** 71 113
2	2 **STX** 2 2	22 **DC2** 12 18	42 2 **”** 22 34	62 18 **2** 32 50	102 2 **B** 42 66	122 18 **R** 52 82	142 2 **b** 62 98	162 18 **r** 72 114
3	3 **ETX** 3 3	23 **DC3** 13 19	43 3 **#** 23 35	63 19 **3** 33 51	103 3 **C** 43 67	123 19 **S** 53 83	143 3 **c** 63 99	163 19 **s** 73 115
4	4 SDC **EOT** 4 4	24 DCL **DC4** 14 20	44 4 **$** 24 36	64 20 **4** 34 52	104 4 **D** 44 68	124 20 **T** 54 84	144 4 **d** 64 100	164 20 **t** 74 116
5	5 PPC **ENQ** 5 5	25 PPU **NAK** 15 21	45 5 **%** 25 37	65 21 **5** 35 53	105 5 **E** 45 69	125 21 **U** 55 85	145 5 **e** 65 101	165 21 **u** 75 117
6	6 **ACK** 6 6	26 **SYN** 16 22	46 6 **&** 26 38	66 22 **6** 36 54	106 6 **F** 46 70	126 22 **V** 56 86	146 6 **f** 66 102	166 22 **v** 76 118
7	7 **BEL** 7 7	27 **ETB** 17 23	47 7 **’** 27 39	67 23 **7** 37 55	107 7 **G** 47 71	127 23 **W** 57 87	147 7 **g** 67 103	167 23 **w** 77 119
8	10 GET **BS** 8 8	30 SPE **CAN** 18 24	50 8 **(** 28 40	70 24 **8** 38 56	110 8 **H** 48 72	130 24 **X** 58 88	150 8 **h** 68 104	170 24 **x** 78 120
9	11 TCT **HT** 9 9	31 SPD **EM** 19 25	51 9 **)** 29 41	71 25 **9** 39 57	111 9 **I** 49 73	131 25 **Y** 59 89	151 9 **i** 69 105	171 25 **y** 79 121
A	12 **LF** A 10	32 **SUB** 1A 26	52 10 ***** 2A 42	72 26 **:** 3A 58	112 10 **J** 4A 74	132 26 **Z** 5A 90	152 10 **j** 6A 106	172 26 **z** 7A 122
B	13 **VT** B 11	33 **ESC** 1B 27	53 11 **+** 2B 43	73 27 **;** 3B 59	113 11 **K** 4B 75	133 27 **[** 5B 91	153 11 **k** 6B 107	173 27 **{** 7B 123
C	14 **FF** C 12	34 **FS** 1C 28	54 12 **,** 2C 44	74 28 **<** 3C 60	114 12 **L** 4C 76	134 28 **** 5C 92	154 12 **l** 6C 108	174 28 **|** 7C 124
D	15 **CR** D 13	35 **GS** 1D 29	55 13 **-** 2D 45	75 29 **=** 3D 61	115 13 **M** 4D 77	135 29 **]** 5D 93	155 13 **m** 6D 109	175 29 **}** 7D 125
E	16 **SO** E 14	36 **RS** 1E 30	56 14 **.** 2E 46	76 30 **>** 3E 62	116 14 **N** 4E 78	136 30 **^** 5E 94	156 14 **n** 6E 110	176 30 **~** 7E 126
F	17 **SI** F 15	37 **US** 1F 31	57 15 **/** 2F 47	77 UNL **?** 3F 63	117 15 **O** 4F 79	137 UNT **_** 5F 95	157 15 **o** 6F 111	177 **DEL (RUBOUT)** 7F 127
	Address Command	**Universal Command**	**Listener Address**		**Talker Address**		**Secondary Command**	

Example



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.	4-34, 4-36
752	Zone edit error	Verify the zone editing conditions	4-36 to 4-38

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

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z