

**DL750**

ScopeCorder

Communication Interface

**U S E R ' S M A N U A L**

---

---

## Foreward

Thank you for purchasing YOKOGAWA's DL750 ScopeCorder.

This Communication Interface User's Manual describes the functions and commands of the GP-IB, serial (RS-232, option), USB, and Network (Ethernet, option) interfaces. To ensure proper use of the GP-IB/serial (RS-232)/USB interfaces, please read this manual thoroughly.

Keep the manual in a safe place for quick reference whenever a question arises.

Two manuals are provided with the DL750 including this Communication Interface User's Manual.

Manual Name	Manual No.	Description
DL750 User's Manual	IM 701210-01E	Describes all functions except for the communications functions and operation procedures of the instrument.
DL750 Communication User's Manual	IM 701210-17E	Describes the communications functions of the GP-IB/serial (RS-232, option)/USB/Network interface.

## Notes

- The contents of this manual are subject to change without prior notice as a result of improvements in 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 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.
- A guarantee card is attached to the instrument. The card will not be reissued, so please read it carefully and keep it in a safe place.
- The TCP/IP software used in this product and the documentation for that TCP/IP software are based in part on BSD Networking Software, Release 1 licensed from The Regents of the University of California.

## USB Interface and Network Interface

- The following items must be installed on the PC to access the communications functions using the USB interface.
  1. A dedicated library for the USB interface.
  2. A USB device driver for the PC-to-DL750 connection.
- The following item must be installed on the PC to access the communications functions using the network interface.
  1. A dedicated library for the network interface.

For information on obtaining the library or the driver, please contact your sales representative.

---

## Trademarks

- MS-DOS is registered trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe, Acrobat, and PostScript are either trademarks or registered trademarks of Adobe Systems incorporated.
- Zip is registered trademarks or trademarks of Iomega corporation in the United States and /or other countries.
- For purposes of this manual, the TM and ® symbols do not accompany their respective trademark names or registered trademark names.
- Other product names are trademarks or registered trademarks of their respective holders.

## Revisions

1st Edition: August, 2002

# How to Use this Manual

## Structure of this Manual

This User's Manual consists of eight 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 interface.

**Chapter 2 Overview of the Serial (RS-232) Interface**

Describes the functions and specifications of serial interface (RS-232).

**Chapter 3 Overview of the USB Interface**

Describes the functions and specifications of USB interface.

**Chapter 4 Overview of the Network Interface (Option)**

Describes the functions and specifications of Network interface.

**Chapter 5 Before Programming**

Describes formats used when sending a command.

**Chapter 6 Commands**

Describes each command.

**Chapter 7 Status Report**

Describes the status byte, various registers and queues.

**Chapter 8 Sample Programs**

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 kS/s (sample rate)
	K	1024 e.g.: 720 KB (floppy disk memory capacity)
Notes	<b>Note</b>	Provides information that is necessary for proper operation of the instrument.
Key	"Probe"	Indicates text of the soft key menu displayed on the screen.

### Symbols Used in Syntax Descriptions

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

Symbol	Description	Example	Example of Input
<x>	Defined value	CHANnel <x> <x>=1 to 4	-> CHANNEL2
{ }	One of the options in { } is selected.	COUPLing {AC DC GND}	-> COUPLING AC
	Exclusive OR		
[ ]	Abbreviated	TRIGger [:SIMPlE]:SLOPe	-> TRIGger:SLOPe

---

# Contents

Foreward .....	i
How to Use this Manual .....	iii
<b>Chapter 1 Overview of the GP-IB Interface</b>	
1.1 Names of the Parts and their Function .....	1-1
1.2 Connection via the GP-IB Interface .....	1-2
1.3 GP-IB Interface Functions .....	1-3
1.4 GP-IB Interface Specifications .....	1-4
1.5 Setting up this Instrument (GP-IB) .....	1-5
1.6 Response to Interface Messages .....	1-6
<b>Chapter 2 Overview of the Serial (RS-232) Interface</b>	
2.1 Names of the Parts and their Function .....	2-1
2.2 Serial (RS-232) Interface Functions and Specifications .....	2-2
2.3 Connection via the Serial (RS-232) Interface .....	2-3
2.4 Handshaking .....	2-5
2.5 Matching the Data Format .....	2-7
2.6 Setting up this Instrument (Serial) .....	2-8
<b>Chapter 3 USB Interface</b>	
3.1 Names of the Parts and their Function .....	3-1
3.2 USB Interface Functions and Specifications .....	3-2
3.3 Connection via the USB Interface .....	3-3
3.4 Setting up this Instrument (USB) .....	3-4
<b>Chapter 4 Network Interface (Option)</b>	
4.1 Names of the Parts and their Function .....	4-1
4.2 Network Interface Functions and Specifications .....	4-2
4.3 Connection via the Network Interface .....	4-4
4.4 Setting up this Instrument (Network) .....	4-5
<b>Chapter 5 Before Programming</b>	
5.1 Messages .....	5-1
5.2 Commands .....	5-3
5.3 Response .....	5-4
5.4 Data .....	5-5
5.5 Synchronization with the Controller .....	5-7
<b>Chapter 6 Commands</b>	
6.1 A List of Commands .....	6-1
6.2 ACQUIRE Group .....	6-14
6.3 ASETUP Group .....	6-16
6.4 CALIBRATE Group .....	6-17
6.5 CAPTURE Group .....	6-18
6.6 CHANNEL Group .....	6-20
6.7 CLEAR Group .....	6-26
6.8 COMMUNICATE Group .....	6-27
6.9 CURSOR Group .....	6-29
6.10 DISPLAY Group .....	6-38

6.11	EVENT Group .....	6-43
6.12	FILE Group .....	6-44
6.13	GONogo Group .....	6-49
6.14	HCOPY Group .....	6-57
6.15	HISTORY Group .....	6-61
6.16	IMAGE Group .....	6-64
6.17	INITialize Group .....	6-66
6.18	LOGic Group .....	6-67
6.19	LSTart Group .....	6-69
6.20	MATH Group .....	6-70
6.21	MEASure Group .....	6-76
6.22	SEARCh Group .....	6-89
6.23	SNAP Group .....	6-93
6.24	SSTart Group .....	6-93
6.25	STARt Group .....	6-93
6.26	STATus Group .....	6-94
6.27	STOP Group .....	6-95
6.28	SYSTem Group .....	6-96
6.29	TIMEbase Group .....	6-100
6.30	TRIGger Group .....	6-101
6.31	WAVEform Group .....	6-116
6.32	XY Group .....	6-120
6.33	ZOOM Group .....	6-122
6.34	Common Command Group .....	6-125

**Chapter 7 Status Report**

7.1	Overview of the Status Report .....	7-1
7.2	Status Byte .....	7-2
7.3	Standard Event Register .....	7-3
7.4	Extended Event Register .....	7-4
7.5	Output Queue and Error Queue .....	7-5

**Chapter 8 Sample Program**

8.1	Before Programming .....	8-1
8.2	Image of Sample Program .....	8-2
8.3	Initialize/Error/Execute .....	8-3
8.4	Sets/Queries the T/Div .....	8-6
8.5	Data Output in Word Format (2 bytes) .....	8-8
8.6	Sets/Queries Measure Value .....	8-14

**Appendix**

Appendix 1	ASCII Character Code .....	App-1
Appendix 2	Error Messages .....	App-2
Appendix 3	Overview of IEEE 488.2-1992 .....	App-4

**Index**

1

2

3

4

5

6

7

8

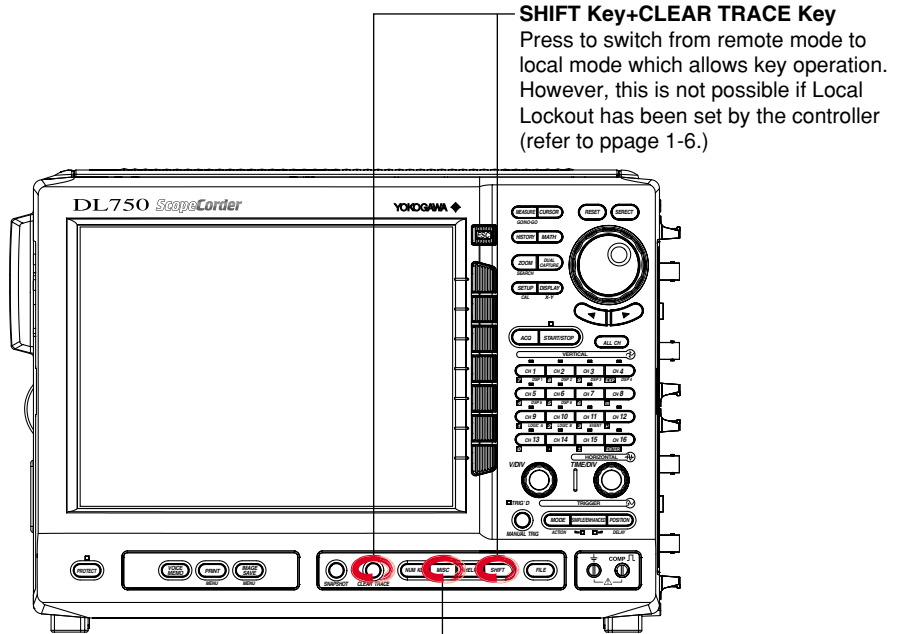
App

Index

# Chapter 1 Overview of the GP-IB Interface

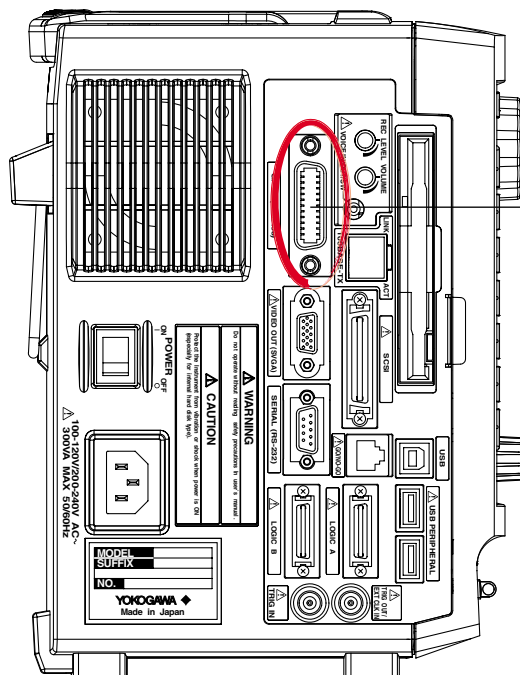
## 1.1 Names of the Parts and their Function

### Front Panel



**MISC Key (Page 1-5)**  
Press to enter the communication settings such as the address and the timeout.

### Side Panel



**GP-IB Connector**  
This connector is for connecting the controller (such as a PC) with the GP-IB cable. For information on how to connect the GP-IB cable, refer to the following page.

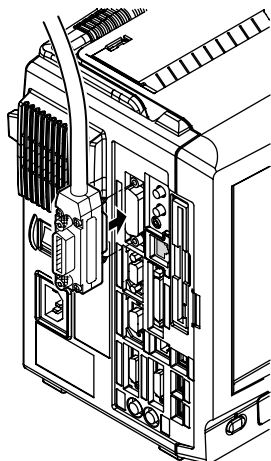
## 1.2 Connection via the GP-IB Interface

### GP-IB Cable

The GP-IB connector on the side panel of the DL750 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.

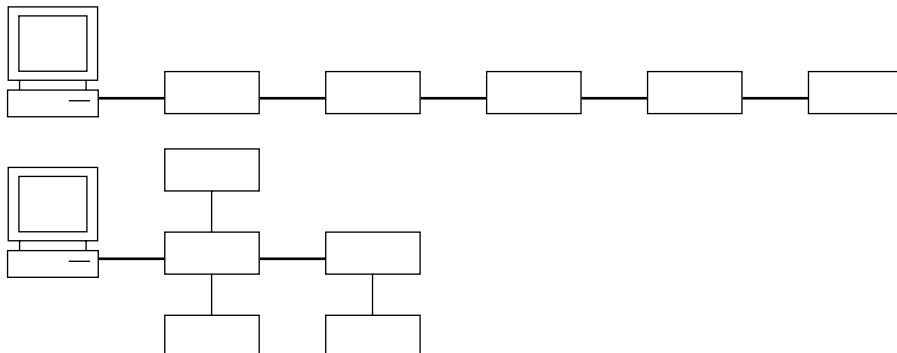
### Connection Method

Connect the GP-IB cable as shown below.



### Connection Precautions

- 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 (e.g. a personal computer) if more than one GP-IB cable is used. 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 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 equipment items must be turned ON.
- When connecting more than one item of equipment, connect them so that the connection route forms a star or linear configuration. Loop or parallel wiring is not allowed.



---

### **CAUTION**

Be sure to switch off power to both your PC and the oscilloscope before connecting or disconnecting cables. Failure to switch power off may cause internal circuit failure or improper operation.

---



## 1.3 GP-IB Interface Functions

### GP-IB Interface Functions

#### Listener Function

- Allows you to make the 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.

#### Note

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

### Switching between Remote and Local Mode

#### When Switched 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.

- REMOTE is displayed on the upper right of the screen.
- All front panel keys except **SHIFT+CLEAR TRACE** can no longer be operated any more.
- Settings entered in local mode are retained.

#### When Switched from Remote to Local Mode

Pressing **SHIFT+CLEAR TRACE** in remote mode puts the instrument in local mode. However, this is not possible if Local Lockout has been set by the controller (page 1-7).

- The REMOTE indicator on the upper right of the screen is turned off.
- All front panel keys are operative.
- Settings entered in remote mode are retained.

#### Note

The GP-IB interface cannot be used simultaneously with other communications interfaces (serial (RS-232), USB, or network).

## 1.4 GP-IB Interface Specifications

### GP-IB Interface Specifications

Electrical and Mechanical Specifications:	Conforms to IEEE Standard 488-1978.
Interface Functions:	Refer to the table below.
Protocol:	Conforms to IEEE Standard 488.2-1992.
Code:	ISO (ASCII) code
Mode:	Addressable mode
Address Setting:	Addresses 0 to 30 can be selected from the GP-IB setting screen, displayed when you press the MISC key.
Remote Mode Clear:	Remote mode can be cleared by pressing <b>SHIFT</b> key+ <b>CLEAR TRACE</b> key. However, this is not possible if Local Lockout has been set by the controller.

#### Interface Functions

Function	Subset Name	Description
Source Handshaking	SH1	Full source handshaking capability
Acceptor Handshaking	AH1	Full acceptor handshaking capability
Talker	T6	Basic talker capability, serial polling, untalk on MLA (My Listen Address), no 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

### Data Transfer Rate

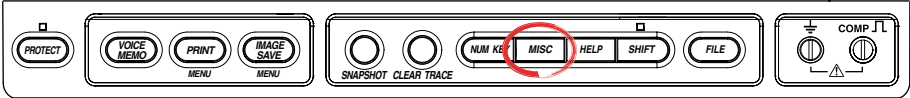
The table below shows approximate response times for output of waveform data, assuming the following configuration.

Model:	701210
Controller:	EPSON Endeavor VZ-6000 (Pentium III, 733 MHz)
GB-IB Board:	NI AT-GPIB/TNT(PNP)
Programming Language:	Visual C++

Data Volume	Byte Format	Word Format	ASCII Format
1000	approx. 64 ms	approx. 68 ms	approx. 530 ms
10000	approx. 114 ms	approx. 160 ms	approx. 5.3 s
100000	approx. 673 ms	approx. 1.2 s	approx. 52.8 s
1000000	approx. 6.5 s	approx. 11.8 s	approx. 530 s

# 1.5 Setting up this Instrument (GP-IB)

## Operating Procedure



### Selecting the Communication Interface

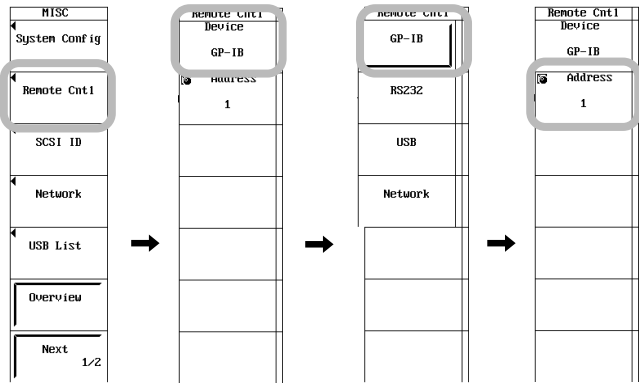
1. Press **MISC**.
2. Press the **Remote Cntl** soft key.
3. Press the **Device** soft key to select GP-IB.

### Note

Only the communications interfaces selected under Device are available. If commands are sent using an unselected communications interface, the command will not be received.

### Setting the Address

4. Turn the **jog shuttle** to set the desired address.



### Explanation

When you make settings which can be made using the front panel keys of the instrument or when you output setup data or waveform data using the controller, the following settings must be made.

### Setting the Address

This function allows you to set the instrument's address for addressable mode within the range of 0 to 30. Each item of equipment connected via a GP-IB interface has its own address, by which it can be identified. Care must be taken to ensure that all interconnected devices are assigned unique addresses.

### Note

Do not change the address while the GP-IB interface is being used by the controller.

## 1.6 Response to Interface Messages

### Response to Interface Messages

#### **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.

#### **Response 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 clears the output queue (page 7-5).

\*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.

#### **Response to a Multi-Line Message (Universal Command)**

##### **LLO(Local Lockout)**

Invalidates **SHIFT+CLEAR** on the front panel to disable 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 performs polling on 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.

### What is an Interface Message?

An interface message is also called an interface command or bus command, and 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 Message**

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

**Address Commands**

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

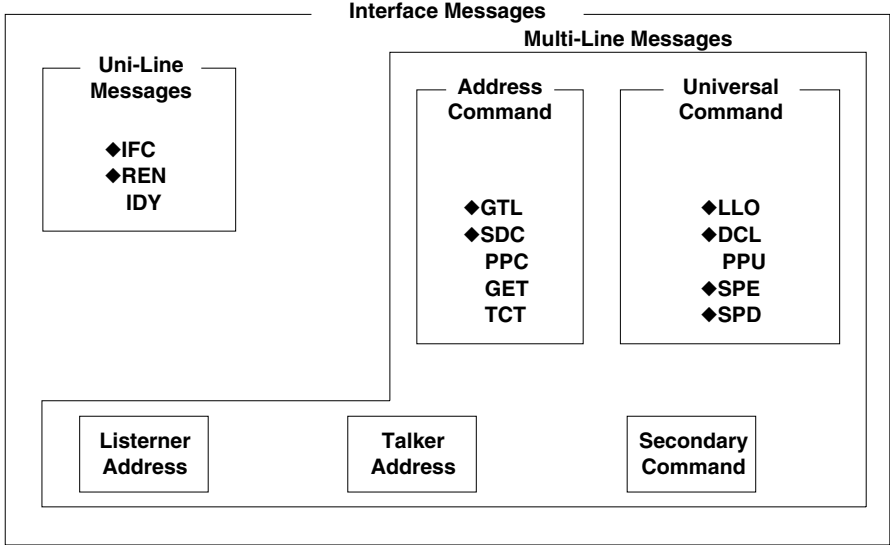
- **Commands valid for equipment designated as a listener;**
  - GTL (Go To Local)
  - SDC (Selected Device Clear)
  - PPC (Parallel Poll Configure)
  - GET (Group Execute Trigger)
- **Command valid for equipment designated as a talker;**
  - TCT (Take ConTrol)

**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 on secondary command can be sent in an interface message.



Messages marked with a “◆” are interface messages supported by the DL750.

**Note**

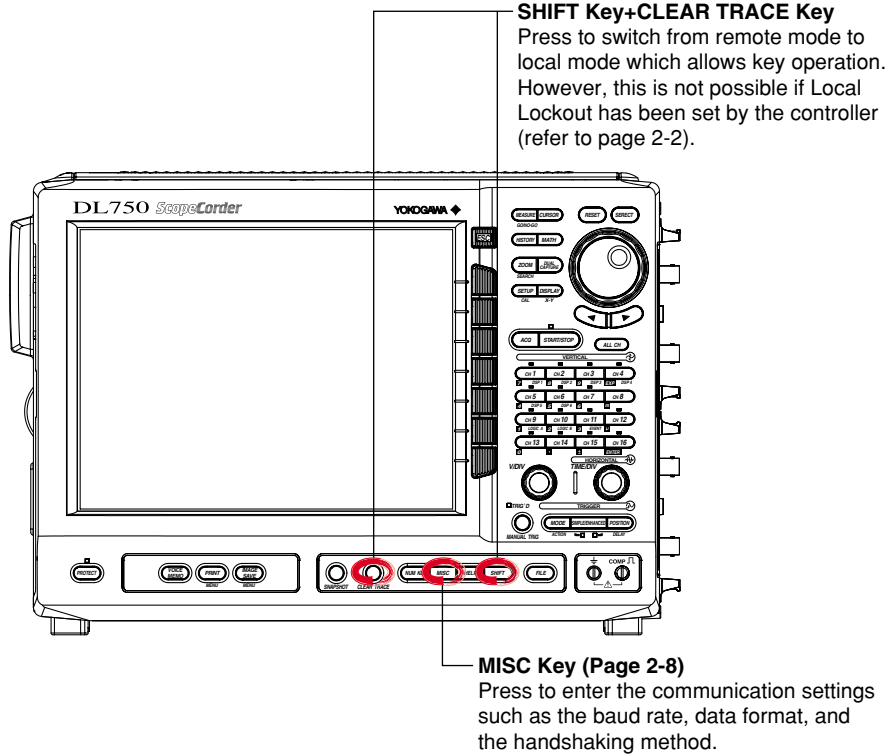
**Differences between SDC and DCL**

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

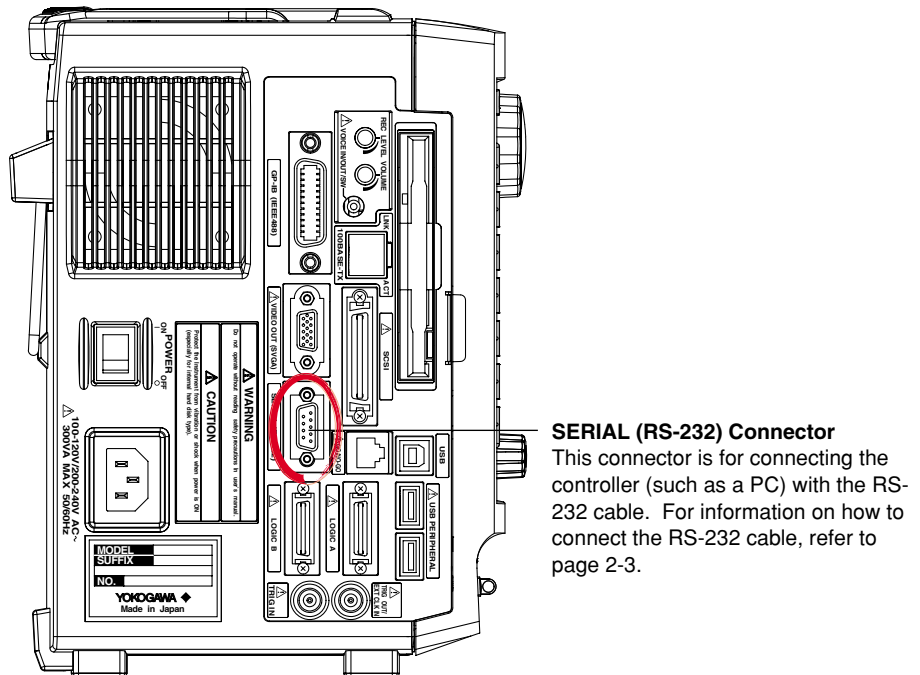
# Chapter 2 Overview of the Serial (RS-232) Interface

## 2.1 Names of the Parts and their Function

### Front Panel



### Side Panel



## 2.2 Serial (RS-232) Interface Functions and Specifications

### Receiving Function

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

Measured/computed data, setup data and error codes can be received.

### Sending Function

Measured/computed data can be output.

Setup data and the status byte can be output.

Error codes which have occurred can be output.

### Serial (RS-232) Interface Specifications

Electrical Characteristics:	Conforms to EIA 574 Standard (EIA-232 (RS-232) Standard for 9-pin)
Connection:	Point-to-point
Communications:	Full-duplex
Synchronization:	Start-stop system
Baud Rate:	1200, 2400, 4800, 9600, 19200, 38400, 57600
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 control both transmission and reception using X-on and X-off signals. (X-on (ASCII 11H), X-off (ASCII 13H))
Receive Buffer Size:	256 bytes

### Switching between Remote and Local Mode

#### When Switched from Local to Remote Mode

Remote mode is activated when the ":COMMunicate:REMote ON" command is received from a controller while local mode is active.

- REMOTE is displayed on the upper right of the screen.
- All front panel keys except **SHIFT+CLEAR TRACE** can no longer be operated any more.
- Settings entered in local mode are retained.

#### When Switched from Remote to Local Mode

Pressing **SHIFT+CLEAR TRACE** in remote mode puts the instrument in local mode.

However, this is not possible if Local Lockout (when the ":COMMunicate:LOCKout ON" command is received) has been set by the controller.

Local mode is activated when the ":COMMunicate:REMote OFF" command regardless of Local Lockout.

- The REMOTE indicator on the upper right of the screen is turned off.
- All front panel keys are operative.
- Settings entered in remote mode are retained.

#### **Note**

The serial interface cannot be used simultaneously with other communications interface (GP-IB, USB, or network).

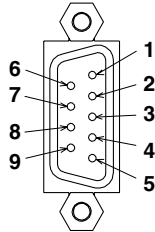
---

## 2.3 Connection via the Serial (RS-232) Interface

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.

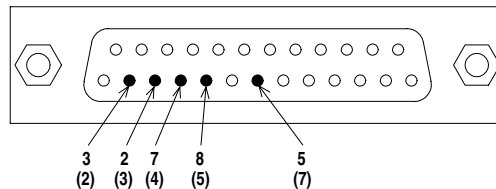
### Connector and Signal Names



- |   |                       |   |
|---|-----------------------|---|
| 2 | RD (Received Data):   | Data received from personal computer.<br>Signal direction...Input.                                      |
| 3 | SD (Send Data):       | Data transmitted to a personal computer.<br>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.<br>Signal direction...Output. |
| 8 | CS (Clear to Send):   | Signal used for handshaking when transmitting data to a personal computer.<br>Signal direction...Input. |

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

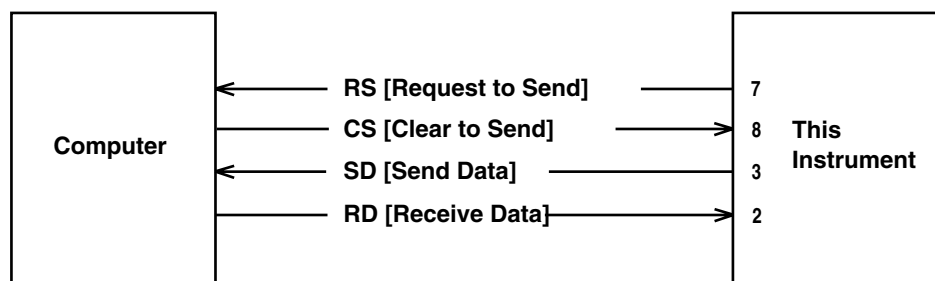
### 9-25 Pin Connector



The number between brackets refer to the pin No. of the 25-pin connector.

### Signal Derection

The figure below shows the direction of the signals used by the serial (RS-232) interface.





### 2.3 Connection via the Serial (RS-232) Interface

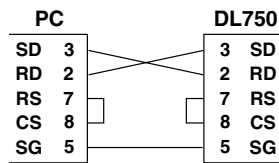
**Table of RS-232 Standard Signals and their JIS and CCITT Abbreviations**

Pin No. (9-Pin Connector)	Abbreviation			Description
	RS-232	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

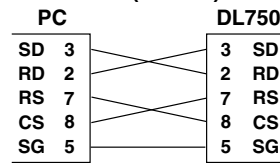
### Signal Line Connection Examples

The pin numbers shown are that of 9-pin connectors.  
In general, use a cross cable.

• **OFF-OFF / XON-XON**



• **CTS-RTS(CS-RS)**



## 2.4 Handshaking

To use an serial (RS-232) 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 three modes.

### Handshake Format Descriptions → ○

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

The Menu of this Instrument

### OFF-OFF

#### Transmission Data Control

There is no handshake between the instrument and host computer. The X-OFF and X-ON 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).

### XON-XON

#### Transmission Data Control

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

#### Reception Data Control

A software handshake status is established between the instrument and host computer. When the instruments reception buffer vacancy reaches 64 bytes, 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).

**CS-RS**

**Transmission Data Control**

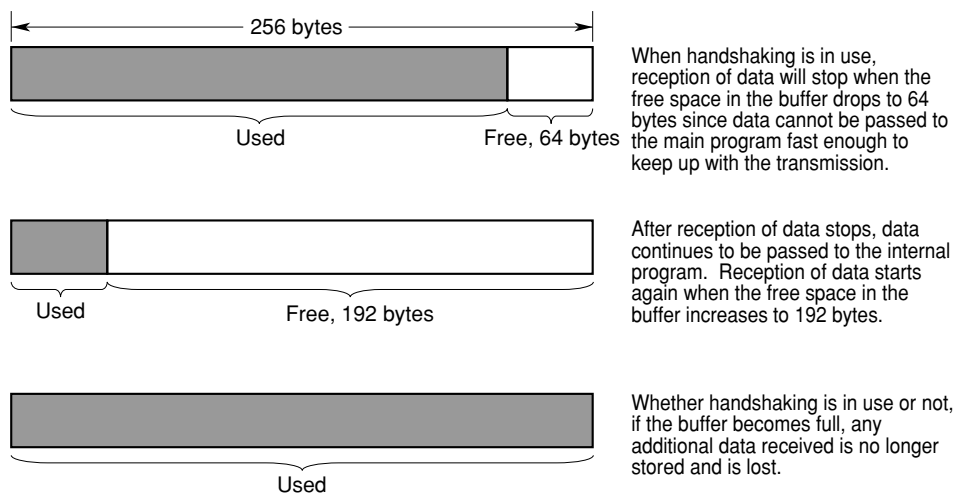
A software handshake status is established between the instrument and host computer. The instrument will stop a data transmission if a “CS = False” status is established, and will resume the transmission when a “CS = True” status is established. The X-OFF and X-ON signals from the host computer are processed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

**Reception Data Control**

A hardware handshake status is established between the instrument and host computer. When the instruments reception buffer vacancy reaches 64 bytes, 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.



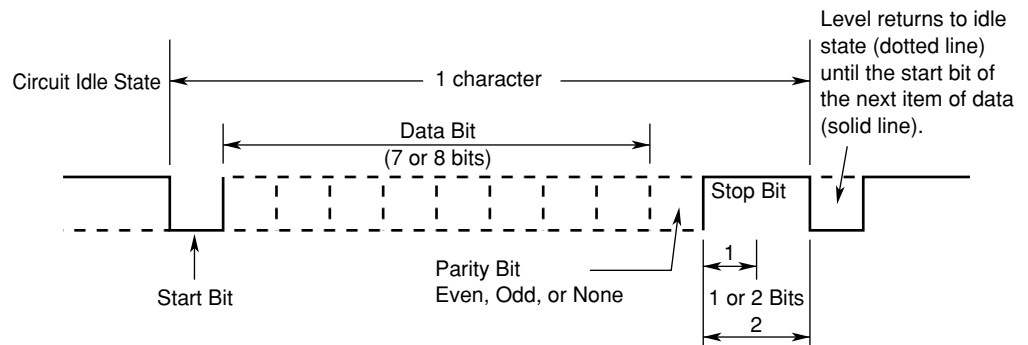
Data Receiving Control Using Handshaking

**Note**

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

## 2.5 Matching the Data Format

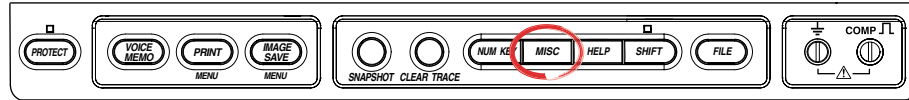
The serial (RS-232) interface of this instrument performs communications using start-stop synchronization. In start-stop synchronization, one character is transmitted at a time. Each character consists of a start bit, data bits, a parity bit, and a stop bit. Refer to the figure below.



## 2.6 Setting up this Instrument (Serial)

### 2.6 Setting up this Instrument (Serial)

#### Operating Procedure



#### Selecting the Communication Interface

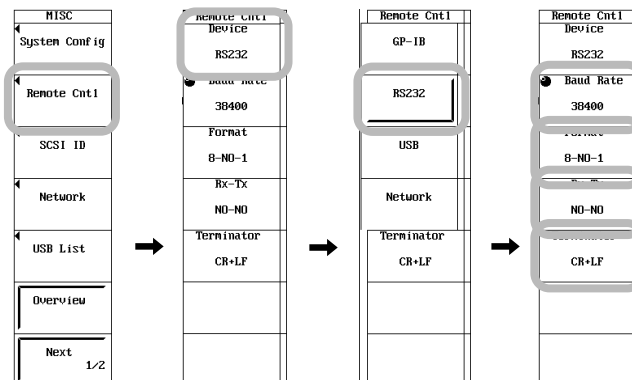
1. Press **MISC**.
2. Press the **Remote Cntl** soft key.
3. Press the **Device** soft key to select RS232.

#### Note

Only the communications interfaces selected under Device are available. If commands are sent using an unselected communications interface, the command will not be received.

#### Selecting the Baud Rate, the Data Format and etc.

4. Press the **Baud Rate (baud rate)**, **Format (data format)**, **Rx-Tx (handshaking method)**, and the **Terminator (terminator)** soft keys individually, and turn the **jog shuttle** to set each item.



**Explanation**

When using the controller to set the items which can be set locally using the keys on the instrument, or when outputting the setup data or the waveform data to the controller, set the following items.

**Baud Rate**

Select from the following choices.

1200, 2400, 4800, 9600, 19200, 38400, 57600

**Data Format**

Select the combination of the data length and the stop bit from the following choices.

8-NO-1, 7-EVEN-1, 7-ODD-1, 7-NO-2

**Handshaking Method**

Select the transmit data control and the receive data control from the following choices.

NO-NO, XON-XON, CTS-RTS

**Terminator**

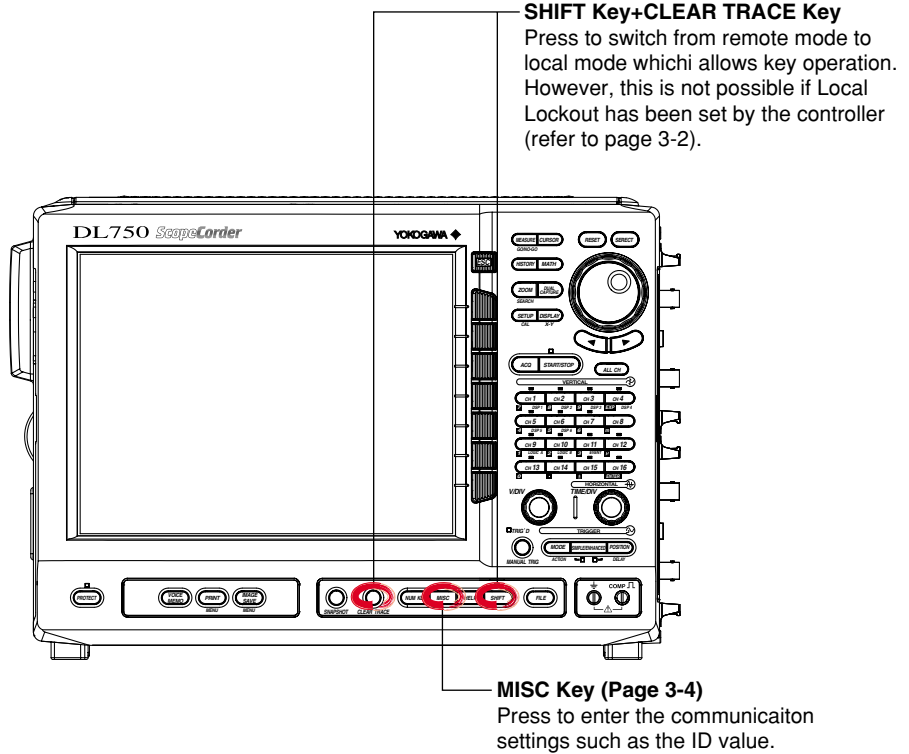
Select from the following choices. The terminator used when sending the data from this instrument is selected on the menu. Use either "LF" or "CR+LF" for the terminator in receiving the data.

CR, LF, CR+LF

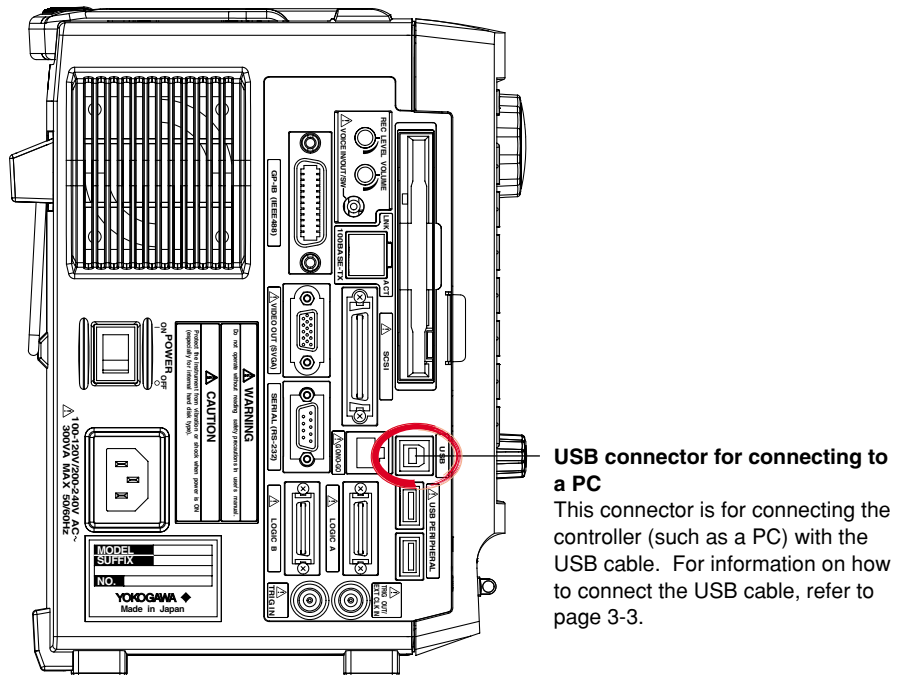
# Chapter 3 USB Interface

## 3.1 Names of the Parts and their Function

### Front Panel



### Side Panel



### 3.2 USB Interface Functions and Specifications

#### Reception Functions

You can specify the same settings as those specified by front panel key operations.  
Receives output requests for measured and computed data, setup data, and error codes.

#### Transmission Functions

Outputs measured and computed data.  
Outputs setting parameters of the panel and the status byte.  
Outputs error codes that are generated.

#### USB Interface Specifications

Electrical and Mechanical Specifications:	Conforms to USB Rev.1.0
Connector:	Type B connector (receptacle)
Number of Ports:	1
Power Supply:	Self-powered
Supported Systems:	PC system supported. Models with a standard USB port that operates on Windows 98 SE, Windows Me, or Windows 2000 (a separate device driver is required for connection to the PC).

#### Switching between Remote and Local Mode

##### When Switching from Local to Remote Mode

If the DL750 receives a ":COMMunicate:REMOte ON" command from the PC when it is in the local mode, it switches to the remote mode.

- REMOTE is displayed at the lower left corner of the screen.
- All keys except **SHIFT+CLEAR TRACE** are disabled.
- The settings that existed in the local mode are maintained even when the DL750 switches to the remote mode.

##### When Switching from Remote to Local Mode

Pressing **SHIFT+CLEAR TRACE** when the instrument is in the remote mode causes the instrument to switch to the local mode. However, this is void when the DL750 has received a ":COMMunicate:LOCKout ON" command from the PC (local lockout condition). When the DL750 receives a ":COMMunicate:REMOte OFF" command from the PC, the DL750 switches to the local mode regardless of the local lock condition.

- REMOTE indication at the lower right corner disappears.
- Key operations are enabled.
- The settings that existed in the remote mode are maintained even when the instrument switches to the local mode.

##### **Note**

---

The USB interface cannot be used simultaneously with other communications interface (GP-IB, serial (RS-232), or network).

---



### **3.3 Connection via the USB Interface**

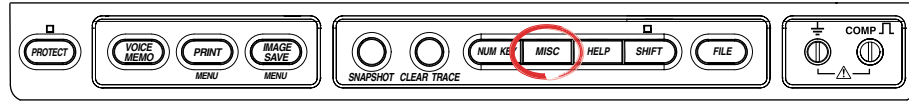
#### **Connection Precautions**

- Connect the USB cable by inserting the connector firmly into the USB connector.
- When connecting multiple devices using USB hubs, connect the DL750 to the USB hub that is closest to the controller.
- Do not mistakenly insert the USB cable to the GO/NO-GO output terminal. This may cause damage to the DL750.
- Do not connect or disconnect the USB cable after the power is turned ON until the DL750 boots up completely (until the DL750 is ready for operation, approximately 20 to 30 s). Doing so can cause damage to the DL750.

### 3.4 Setting up this Instrument (USB)

## 3.4 Setting up this Instrument (USB)

### Operating Procedure



#### Selecting the Communication Interface

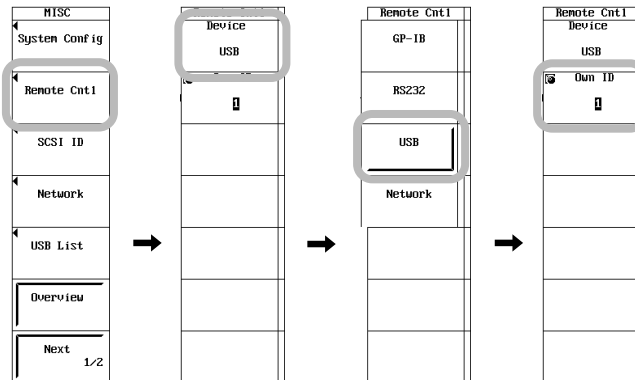
1. Press **MISC**.
2. Press the **Remote Cntl** soft key.
3. Press the **Device** soft key to select USB.

#### Note

Only the communications interfaces selected under Device are available. If commands are sent using an unselected communications interface, the command will not be received.

#### Setting the ID Value

4. Turn the **jog shuttle** to set the ID value.



### Explanation

Carry out the following settings when using a controller to set information that can be specified through key operation on the DL750 or when outputting setup data or output waveform data to the controller.

#### Setting the ID Value

Set the ID value of the DL750 within the following range:

1 to 127

You can connect multiple devices to a controller using the USB interface. If multiple devices are connected in a single USB system, the ID value is used by the controller to identify each device. Therefore, unique ID values must be assigned to the devices within a single system.

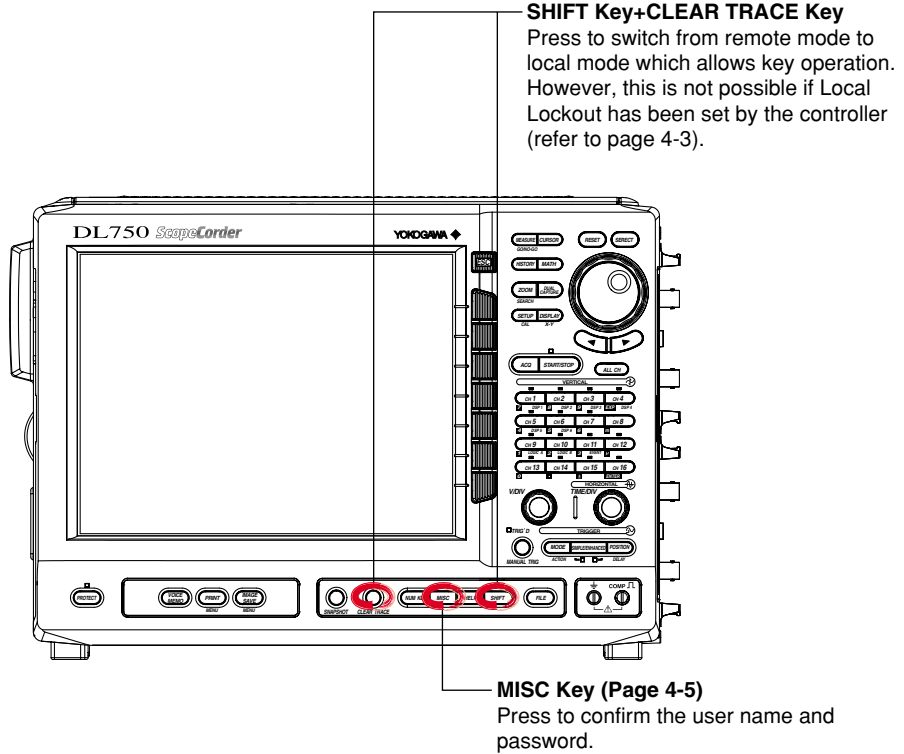
#### Note

Do not change the ID value while using the USB.

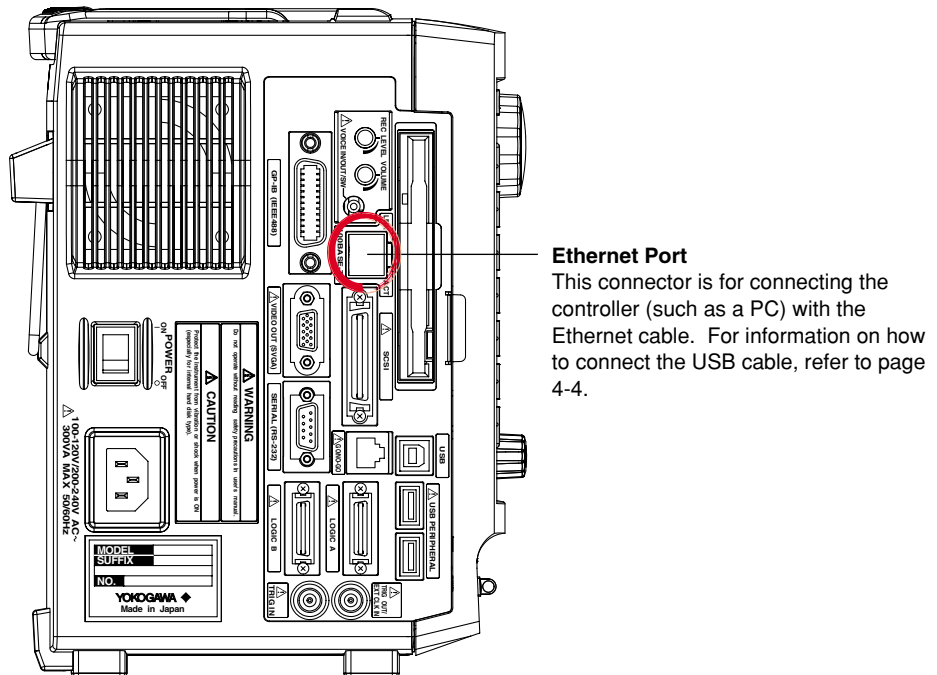
# Chapter 4 Network Interface (Option)

## 4.1 Names of the Parts and their Function

### Front Panel



### Side Panel



### 4.2 Network Interface Functions and Specifications

#### Receiving Function

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

Measured or computed data, setup data, and error codes can be received.

#### Sending Function

Measured/computed data can be output.

Setup data and the status byte can be output.

Error codes which have occurred can be output.

#### Network Interface Specifications

Communication Port Number:	1
Electrical and Mechanical Specifications:	IEEE802.3 Compliant
No. of simultaneous connections:	1
Transmission Method:	100BASE-TX/10BASE-T
Transmission Rate:	Max. 100 Mbps
Communication Protocol:	TCP/IP
Connector Type:	RJ-45 connector
Port No.:	10001/tcp

#### Data Transfer Rate

The table below shows approximate response times for output of waveform data, assuming the following configuration.

Model:	701210
Controller:	EPSON Endeavor VZ-6000 (Pentium III, 733 MHz)
Network Adaptor:	Corega F Ether PCI-TXL
Programming Language:	Visual C++

Data Volume	Byte Format	Word Format	ASCII Format
1000	approx. 9 ms	approx. 9 ms	approx. 464 ms
10000	approx. 25 ms	approx. 31 ms	approx. 3.7 s
100000	approx. 195 ms	approx. 404 ms	approx. 36 s
1000000	approx. 1.9 s	approx. 2.5 s	approx. 380 s

## Switching between Remote and Local Mode

### When Switched from Local to Remote

Remote mode is activated when the ":COMMunicate:REMOte ON" command is received from a controller while local mode is active.

- REMOTE is displayed on the upper right of the screen.
- All front panel keys except **SHIFT +CLEAR TRACE** can no longer be operated.
- Settings entered in local mode are retained even when switching to remote mode.

### When Switched from Remote to Local Mode

Pressing **SHIFT +CLEAR TRACE** in remote mode puts the instrument in local mode. However, this is not possible when the ":COMMunicate:REMOte ON" command is received from the controller while Local Lockout mode is active. Local mode is activated when the ":COMMunicate:REMOte OFF" command is received regardless of Local Lockout.

- The REMOTE indicator on the upper right of the screen is turned off.
- All front panel keys are operative.
- Settings entered in remote mode are retained even when switching to local mode.

### Note

The network interface cannot be used simultaneously with other communications interfaces (GP-IB, serial (RS-232), or USB).

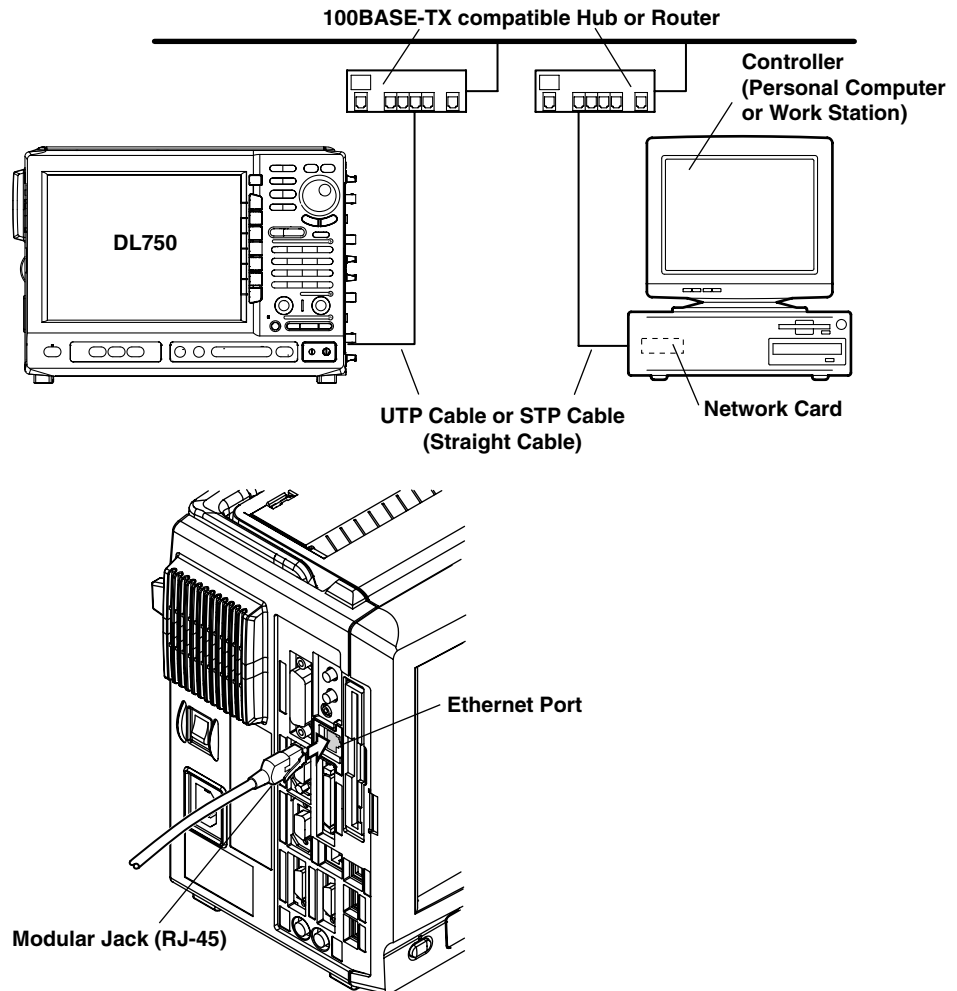
## User Verification Function

The user must enter a user name and password for the interface when connecting to the network. The user name and password is set on the instrument in the Remote Cntl setting screen under the MISC menu. For details, see section 4.4, "Setting up this Instrument (Network)."

## 4.3 Connection via the Network Interface

### Connection Method

Connect a UTP (Unshielded Twisted-Pair) or STP (Shielded Twisted-Pair) cable that has been connected to a hub to the 100BASE-TX port on the rear panel of the unit.

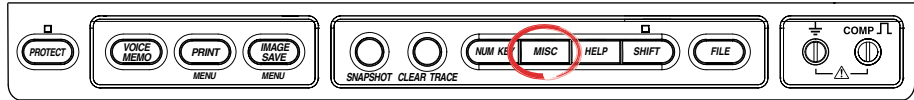


### Connection Precautions

- You must use a straight cable when connecting the unit to a controller or a work station. We cannot guarantee results using a cross cable.
- When using a UTP cable (straight cable), be sure to use only a category 5 cable.

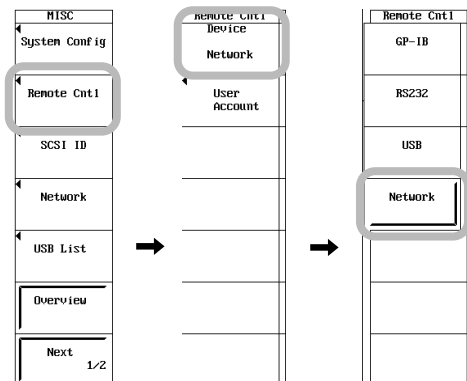
## 4.4 Setting up this Instrument (Network)

### Operating Procedure



#### Selecting the Communication Interface

1. Press **MISC**.
2. Press the **Remote Cntl** soft key.
3. Press the **Device** soft key, and select Network.

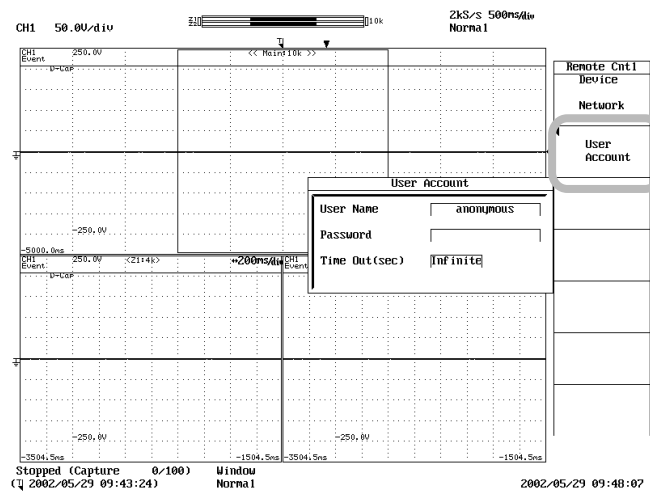


#### Note

Only the communications interfaces selected under Device are available. If commands are sent using an unselected communications interface, the command will not be received.

#### Setting the User Name, Password, and Timeout Time

4. The **User Account** dialog box opens. The User Account dialog box opens.
5. Use the **jog shuttle** to move the cursor to User Name, then press Select to display the keyboard.



## 4.4 Setting up this Instrument (Network)

---

6. Enter the user name. If you do not wish to limit access, use the anonymous setting. If access is to be limited, enter a user name of 15 characters or less.
7. Use the **jog shuttle** to move the cursor to Password, then press **SELECT** to display the keyboard.
8. Enter a password of 15 characters or less. Reenter the password to confirm.
9. Use the **jog shuttle** to move the cursor to Time Out, then press **SELECT** to display the keyboard.
10. Enter the Timeout time using the **jog shuttle**.

### **Note**

---

- For (soft) keyboard operation, see section 3.2, “Entering Values and Strings” in user’s manual IM701210-01E.
  - The user name and password are case sensitive.
- 

### **TCP/IP Settings**

The following TCP/IP settings must be entered in order to use the network interface functions.

- IP Address
- Subnet Mask
- Default Gateway

For details about how to enter these settings, see section 12.2, “Setting up the TCP/IP” in user’s manual IM701210-01E.

## **Description**

Use the settings below to setup communications so that you can input settings on the instrument using the controller, or output settings or values to the controller.

### **Setting the User Name and Password**

There is a user verification function for the Network interface. The user name and password are set on the instrument ahead of time.

- **User Name Setting**  
15 characters or less may be used. The default value is anonymous.
- **Password Setting**  
15 characters or less may be used.

### **Setting the Timeout Time**

If the unit is not being accessed at the specified timeout time, the connection to the network will be automatically closed.

### **TCP/IP Settings**

For details, see section 12.2, “Setting up the TCP/IP” in user’s manual IM701210-01E.

### **Note**

---

- If a user verification error occurs, the connection to the instrument is cut.
  - If the login name is anonymous, it is not necessary to enter a password.
  - When changing the user name, password, timeout time, or TCP/IP settings, the settings will not take effect until the unit is turned OFF then back ON again.
-



# Chapter 5 Before Programming

## 5.1 Messages

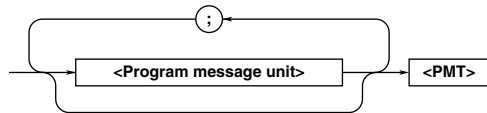
### Messages

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

The format of a program message is shown below.



#### <Program Message Unit>

A program message consists of one 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 :ACQUIRE:MODE NORMAL;COUNT 1<PMT>

Unit                      Unit

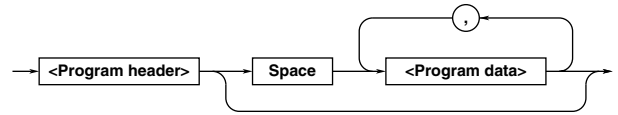
#### <PMT>

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 5-3.

#### <Program Data>

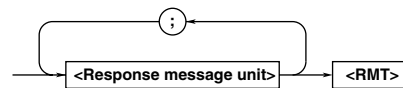
If certain conditions are required for the execution of a command, program data must be added. Program data must be separated from the header by a space (ASCII code "20H"). If multiple items of program data are included, they must be separated by a ", " (comma). For details, refer to page 5-5.

Example :ACQUIRE:MODE NORMAL<PMT>

Header                      Data

### Response Messages

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 :ACQUIRE:MODE NORMAL;COUNT 1<RMT>

Unit                      Unit

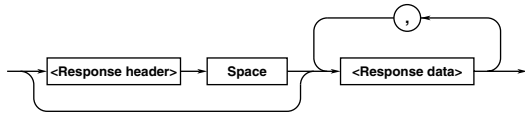
#### <RMT>

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

## 5.1 Messages

### Response Message Unit Format

The format of a program message unit is shown below.



#### <Response Header>

A response header sometimes precedes the response data. Response data must be separated from the header by a space. For details, refer to page 5-4.

#### <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 5-5.

#### Example

1.25E-02<RMT>      :ACQUIRE:MODE NORMAL<RMT>  
Data                                  Header                  Data

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

### Points to Note Concerning Message Transmission

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

### Dead Lock

This instrument has a buffer memory in which both program and response messages of 1024 bytes or more can be stored. (The number of bytes available will vary depending on the operating state of the instrument.) If both buffer memories become full at the same time, this instrument becomes inoperative. This state is called deadlock. 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 deadlock will occur if the program message does not contain a query.

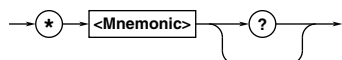
## 5.2 Commands

### 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.

### 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.

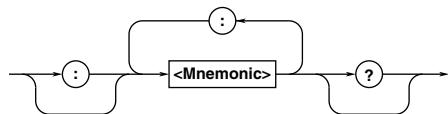


An example of a common command

\*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.

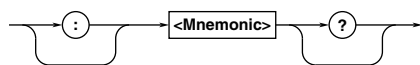


An example of a compound header

:ACQUIRE:MODE

### 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 acquisition settings

```
:ACQUIRE:AVERAGE:COUNT
:ACQUIRE:MODE
:ACQUIRE:AVERAGE:EWEIGHT
:ACQUIRE:CLOCK
:ACQUIRE:RLENGTH
:ACQUIRE:COUNT
```

#### 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:ACQUIRE:MODE NORMAL;COUNT 1<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. It is impossible to omit a colon (:).

Example :ACQUIRE:MODE NORMAL;:DISPLAY:
FORMAT SINGLE<PMT>

#### When Concatenating Simple Headers

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

Example:ACQUIRE:MODE NORMAL;: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:ACQUIRE:MODE NORMAL;\*CLS;
COUNT 1<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:ACQUIRE:MODE
NORMAL<PMT>;ACQUIRE:COUNT 1<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: CHANnel1?<PMT> -> :CHANNEL1:
      DISPLAY ON; LABEL "CH1  ";
      COUPLING DC; POSITION 0.00;
      PROBE 10; VDIV 50.0E+00;
      BWIDTH FULL; OFFSET 0.0E+00; LSCALE:
      MODE 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 "TRIGger[:SIMPLle]:LEVel" can be written as "TRIG:LEV."

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

Example "TRIGger?" and "TRIGger:SIMPLe?" belong to different upper-level query levels.

## 5.3 Response

### 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 :MEASURE:CHANnel1:PTOPeak:
      VALUE?<PMT> -> 10.0E+00<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.

## 5.4 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: Type of probe of CH2 -> CHANnel2:PROBe 100)
<Voltage><Current> <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 -> CHANnel1:COUPling{AC DC GND})
<Boolean>	Indicates ON/OFF. Set to ON, OFF or value (Example: CH2 display ON -> CHANnel2:DISPlay ON)
<Character string data>	Arbitrary character string (Example: Comment on screen-data output -> HCOpy:COMment "ABCDEF")
<Filename>	Gives the name of a file. (Example: Name of file to be saved -> FILE:SAVE:WAVEform:NAME "CASE1")
<Block data>	Arbitrary 8-bit data (Example: Response to acquired waveform data -> #800000010ABCDEFGHIJ)

### <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>, <Current>, <Time>, <Frequency>

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

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

### • <Multiplier>

Multipliers which can be used are shown below.

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

### • <Unit>

Units which can be used are shown below.

Symbol	Word	Description
V	Volt	Voltage
A	Ampere	Current
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.

## 5.4 Data

### <Register>

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

Form	Example
<NRf>	1
#H	#H0F
<Hexadecimal value made up of the digits 0 to 9, and A to F>	
#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 5-4.

Form	Example
{AC DC DC50 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' "IEEE488.2-1992"

- 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.

### <Filename>

Gives the name of a file. The format is as follows.

Form	Example
{<NRf> <Character data> <Character string>}	1 CASE "CASE"

- If you input an <NRf> value, the system converts the value (after rounding to the nearest integer) to the corresponding 8-character ASCII string. (If you set the value to 1, the name becomes "00000001".) Note that negative values are not allowed.
- If you enter a <character data> or <character string> argument that is longer than eight characters, only the first eight characters are used.
- Response messages always return filenames as <character string> arguments.

**<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>	#80000010ABCDEFGHIJ

- #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-digits decimal value>  
Indicates the number of bytes of data. (Example: 0000010=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.

**5.5 Synchronization with the Controller****Overlap Commands and Sequential Commands**

There are two kinds of command; overlap commands and sequential commands. Execution of an overlap command may start before execution of the previously sent command is completed.

The "CHANnel1:VDIV" command, for example, is a sequential command. Assume that you set a new V/div value and immediately request return of the new value, as follows:

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

In this case, the oscilloscope always returns the newest setting ("5V"). This is because it always completes processing of the current sequential command (in this case, "VDIV 5V") before moving on to the next command ("VIDV?").

In contrast, assume that you begin a file load and then immediately query the V/div value:

```
:FILE:LOAD:SETup:EXECute "CASE1"; :  
CHANnel1:VDIV?
```

Because "FILE:LOAD:SETup:EXECute "CASE1"" is an overlapped command, the oscilloscope will advance to the "CHANnel1:VDIV?" command before it finishes the load. The returned V/div value will not show the newest setting, but will rather show the setting in use before the setup was changed.

Obviously, use of overlapped commands may in some cases produce inappropriate results. Where necessary, you can avoid such problems as described below.

**Synchronization with an Overlap Command****Using the \*WAI Command**

The \*WAI command causes the commands which follow it to wait until an overlap command has been executed.

```
Example: COMMunicate:OPSE #0040;:FILE:  
LOAD:SETup:EXECute "CASE1";*WAI; :  
CHANnel1:VDIV?<PMT>
```

The "COMMunicate:OPSE" command is used to designate which commands are to be subject to the \*WAI command. In the above example, only auto set-up is designated.

Since a \*WAI command is executed just before "CHANnel1:VDIV?," "CHANnel1:VDIV?" will not be executed until auto set-up has been completed.

## 5.5 Synchronization with the Controller

---

### Using the COMMunicate:OVERlap Command

The "COMMunicate:OVERlap" command is used to enable or disable overlap operation.

Example: COMMunicate:OVERlap #HFFBF;:FILE:  
LOAD:SETup:EXECute "CASE1";:  
CHANnel1:VDIV:VALue?<PMT>

The "COMMunicate:OVERlap #HFFBF" command disables overlapped operation of the medium access command, while enabling all other overlap-type operations. The oscilloscope will therefore handle "FILE:LOAD:SETup:EXECute "CASE1"" as sequential command, ensuring that the "CHANnel1:VDIV?" command (in the above example) will not execute until file loading is completed.

### Using the \*OPC Command

The \*OPC command causes the OPC bit (bit 0) of the standard event register (page 7-3) to be set to "1" when an overlap operation has been completed.

Example: COMMunicate:OPSE #H0040;\*ESE 1;  
\*ESR?;\*SRE 32;:FILE:LOAD:SETup:  
EXECute "CASE1";\*OPC<PMT>  
(Response to \*ESR? is decoded.)  
(Service request is awaited.)  
CHANnel1:VDIV?<PMT>

The "COMMunicate:OPSE" command is used to designate which commands are to be subject to the \*OPC command. In the above example, only medium access commands are designated.

\*ESE 1 and \*SRE 32 stipulate that a service request is generated only when the OPC bit is set to "1." \*ESR? is used to clear the standard event register. In the above example, "CHANnel1:VDIV?" will not be executed until a service request is generated.

### Using the \*OPC? Query

The \*OPC? query generates a response when an overlap operation has been completed.

Example: COMMunicate:OPSE #H0040;:FILE:  
LOAD:SETup:EXECute "CASE1";  
\*OPC?<PMT>  
(Response to \*OPC? is decoded.)  
:CHANnel1:VDIV?<PMT>

The "COMMunicate:OPSE" command is used to designate which commands are to be subject to the \*OPC? command. In the above example, only medium access commands are designated.

Since \*OPC? does not generate a response until an overlap operation is completed, file loading will have been completed when a response to \*OPC? is read.

### Note

Most commands are sequential commands. Commands used in Chapter 6 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 SINGLE;: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 7-4). 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 SINGLE;: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 7-4).

```
Example : STATus:FILTER1 FALL; : STATus:
        EESE 1; EESR?; *SRE 8; : TRIGger:
        MODE SINGLE; : START<PMT>
        (Response to "STATus:EESR?" is decoded.)
        (Service request is awaited.)
        : WAVEform: SEND?<PMT>
```

The "STATus:FILTER1 FALL" command sets the transition filter such that Bit 0 (FILTER1) of the Extended Event Register sets to 1 when Bit 0 of the Condition Register changes 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:
        EESR?; : TRIGger:MODE SINGLE; :
        START<PMT>
        (Response to "STATus:EESR?" is decoded.)
        : COMMunicate:WAIT 1; : WAVEform:
        SEND?<PMT>
```

For a description of "STATus:FILTER1 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."

# Chapter 6 Commands

## 6.1 A List of Commands

Command	Function	Page
<b>ACQUIRE Group</b>		
:ACQUIRE?	Queries all settings related to the waveform acquisition.	6-15
:ACQUIRE:AVERAGE?	Queries all settings related to averaging.	6-15
:ACQUIRE:AVERAGE:COUNT	Sets the waveform acquisition count or queries the current setting.	6-15
:ACQUIRE:AVERAGE:EWEIGHT	Sets the attenuation constant of exponential averaging or queries the current setting.	6-15
:ACQUIRE:CLOCK	Sets the time base or queries the current setting.	6-15
:ACQUIRE:COUNT	Sets the waveform acquisition count when not in average mode or queries the current setting.	6-15
:ACQUIRE:MODE	Sets the waveform acquisition mode or queries the current setting.	6-15
:ACQUIRE:PROTATE	Sets pulse/rotate when using external clock or queries the current setting.	6-15
:ACQUIRE:RENGTH	Sets the record length or queries the current setting.	6-15
:ACQUIRE:RTOUT?	Queries all settings related to realtime recording.	6-15
:ACQUIRE:RTOUT:MODE	Sets whether to perform realtime recording or queries the current setting.	6-15
:ACQUIRE:RTOUT:ANAMING	Sets auto naming for realtime recording or queries the current setting.	6-15
:ACQUIRE:RTOUT:FILENAME	Sets the file name of auto naming of realtime recording or queries the current setting.	6-16
:ACQUIRE:RTOUT:COMMENT	Sets the comment of realtime recording or queries the current setting.	6-16
<b>ASETUP Group</b>		
:ASETUP?	Queries all settings related to auto setup.	6-16
:ASETUP:ADJUST	Sets the center position after auto setup or queries the current setting.	6-16
:ASETUP:EXECUTE	Executes auto setup.	6-16
:ASETUP:TARGET	Sets the target channel for auto setup or queries the current setting.	6-16
:ASETUP:UNDO	Cancel auto setup.	6-16
<b>CALIBRATE Group</b>		
:CALIBRATE?	Queries all settings related to the calibration.	6-17
:CALIBRATE[:EXECUTE]	Executes the calibration.	6-17
:CALIBRATE:MODE	Sets whether to perform auto calibration or queries the current setting.	6-17
<b>CAPTURE Group</b>		
:CAPTURE?	Queries all settings related to dual capture.	6-19
:CAPTURE:BITDISP	Sets the dual capture event bit or queries the current setting.	6-19
:CAPTURE:CAPNUM	Sets the capture number or queries the current setting.	6-19
:CAPTURE:MAG	Sets the zoom rate of the sub waveform window or queries the current setting.	6-19
:CAPTURE:MODE	Sets whether or not to perform dual capture or queries the current setting.	6-19
:CAPTURE:POSITION	Sets the horizontal position of the sub waveform window or queries the current setting.	6-19
:CAPTURE:TDIV	Sets the T/div value of the capture or queries the current setting.	6-19
:CAPTURE:WINDOW	Sets the size of the sub waveform window or queries the current setting.	6-19
<b>CHANNEL Group</b>		
:CHANNEL<x>?	Queries all settings related to the vertical axis of the channel.	6-22
:CHANNEL<x>:DISPLAY	Turns ON/OFF the waveform display of the channel or queries the current setting.	6-22
:CHANNEL<x>:LABEL	Sets the waveform label of channel or queries the current setting.	6-22
:CHANNEL<x>:MODULE?	Queries the installed modules.	6-22
:CHANNEL<x>:TEMPERATURE?	Queries all settings when the Temperature, High Precision Voltage Isolation Module.	6-23
:CHANNEL<x>:TEMPERATURE:BURNOUT	Sets whether to detect burnouts when the Temperature, High Precision Voltage Isolation Module is installed or queries the current setting.	6-23
:CHANNEL<x>:TEMPERATURE:BWIDTh	Sets the filter when the Temperature, High Precision Voltage Isolation Module is installed or queries the current setting.	6-23
:CHANNEL<x>:TEMPERATURE:COUPLing	Sets the input coupling when the Temperature, High Precision Voltage Isolation Module is installed or queries the current setting.	6-23
:CHANNEL<x>:TEMPERATURE:RJC	Sets the RJC when the Temperature, High Precision Voltage Isolation Module is installed or queries the current setting.	6-23
:CHANNEL<x>:TEMPERATURE:SCALE	Sets the upper and lower limits on the screen when the Temperature, High Precision Voltage Isolation Module is installed or queries the current setting.	6-23

## 6.1 A List of Commands

Command	Function	Page
:CHANnel<x>:TEMPerature:TYPE	Sets the thermocouple to be used when the Temperature, High Precision Voltage Isolation Module is installed or queries the current setting.	6-23
:CHANnel<x>:TEMPerature:UNIT	Sets the unit of the upper and lower limits when the Temperature, High Precision Voltage Isolation Module is installed or queries the current setting.	6-24
:CHANnel<x>:VOLTagE?	Queries all settings when the voltage module* is installed.	6-24
:CHANnel<x>[:VOLTagE]:BWIDth	Sets the filter when the voltage module* is installed or queries the current setting.	6-24
:CHANnel<x>[:VOLTagE]:COUPling	Sets the input coupling when the voltage module* is installed or queries the current setting.	6-24
:CHANnel<x>[:VOLTagE]:INVert	Sets whether to invert the waveform when the voltage module* is installed or queries the current setting.	6-24
:CHANnel<x>[:VOLTagE]:LSCale?	Queries all settings related to linear scaling when the voltage module* is installed.	6-24
:CHANnel<x>[:VOLTagE]:LSCale:AVALue	Sets scaling coefficient A when the voltage module* is installed or queries the current setting.	6-24
:CHANnel<x>[:VOLTagE]:LSCale:BVALue	Sets offset B when the voltage module* is installed or queries the current setting.	6-25
:CHANnel<x>[:VOLTagE]:LSCale:GETMeasure	Executes the measurement of the P1:X P2:X values of linear scaling when the voltage module is installed.	6-25
:CHANnel<x>[:VOLTagE]:LSCale:MODE	Sets the linear scaling function when the voltage module* is installed or queries the current setting.	6-25
:CHANnel<x>[:VOLTagE]:LSCale:{P1X P1Y P2X P2Y}	Sets the P1:X P1:Y P2:X P2:Y values of linear scaling when the voltage module* is installed or queries the current setting.	6-25
:CHANnel<x>[:VOLTagE]:LSCale:UNIT	Sets the unit to be added to the result of linear scaling when the voltage module* is installed or queries the current setting.	6-25
:CHANnel<x>[:VOLTagE]:OFFSet	Sets the offset voltage when the voltage module* is installed or queries the current setting.	6-25
:CHANnel<x>[:VOLTagE]:POSition	Sets the vertical position when the voltage module* is installed or queries the current setting.	6-25
:CHANnel<x>[:VOLTagE]:PROBe	Sets the probe attenuation when the voltage module* is installed or queries the current setting.	6-25
:CHANnel<x>[:VOLTagE]:SCALe	Sets the upper and lower limits on the screen when the voltage module* is installed or queries the current setting.	6-26
:CHANnel<x>[:VOLTagE]:VARiable	Sets the variable when the voltage module* is installed or queries the current setting.	6-26
:CHANnel<x>[:VOLTagE]:VDIV	Sets V/div when the voltage module* is installed or queries the current setting.	6-26
:CHANnel<x>[:VOLTagE]:ZOOM	Sets the vertical zoom rate when the voltage module* is installed or queries the current setting.	6-26
* Voltage module refers to the High-Speed 10 MS/s, 12-Bit Isolation Module and the High-Speed 1 MS/s, 16-Bit Isolation Module.		
<b>CLEar Group</b>		
:CLEar	Executes clear trace.	6-26
<b>COMMunicate Group</b>		
:COMMunicate?	Queries all settings related to communications.	6-27
:COMMunicate:HEADer	Sets whether to attach a header to the response data or queries the current setting.	6-27
:COMMunicate:LOCKout	Sets or clears local lockout.	6-27
:COMMunicate:OPSE	Sets the overlap command that is to be used by the *OPC, *OPC?, and *WAI commands or queries the current setting.	6-28
:COMMunicate:OPSR?	Queries the operation pending status register.	6-28
:COMMunicate:OVERlap	Sets the commands that will operate as overlap commands or queries the current setting.	6-28
:COMMunicate:REMOte	Switches between remote and local.	6-28
:COMMunicate:STATus?	Queries line-specific status.	6-28
:COMMunicate:VERBose	Sets whether to use abbreviated or unabbreviated form for response data.	6-28
:COMMunicate:WAIT	Waits for an extended event to occur.	6-28
:COMMunicate:WAIT?	Creates a response for the specified extended event.	6-28
<b>CURSor Group</b>		
:CURSor?	Queries all settings related to cursor measurements.	6-32
:CURSor[:TY]?	Queries all settings related to the cursor of the T-Y display.	6-32
:CURSor[:TY]:DEGRee?	Queries all settings related to the angle cursor.	6-32
:CURSor[:TY]:DEGRee:DX?	Queries the angle difference !s!DD between angle cursors.	6-32
:CURSor[:TY]:DEGRee:DY?	Queries the !s!DV value between angle cursors.	6-32

Command	Function	Page
:CURSor[:TY]:DEGRee:JUMP	Jumps the angle cursor onto the zoom waveform.	6-32
:CURSor[:TY]:DEGRee:POSition<x>	Sets the angle cursor position or queries the current setting.	6-32
:CURSor[:TY]:DEGRee:REFerence<x>	Sets the position of the start point (Reference 1) or the end point (Reference 2) of the angle reference or queries the current setting.	6-32
:CURSor[:TY]:DEGRee:RVALue	Sets the angle reference width or queries the current setting.	6-32
:CURSor[:TY]:DEGRee:TRACe	Sets the target waveform of the angle cursor or queries the current setting.	6-33
:CURSor[:TY]:DEGRee:X<x>?	Queries the angle of the angle cursor.	6-33
:CURSor[:TY]:DEGRee:Y<x>?	Queries the voltage of the angle cursor.	6-33
:CURSor[:TY]:HORizontal?	Queries all settings related to H cursor.	6-33
:CURSor[:TY]:HORizontal:DY?	Queries the Y-axis value between the H cursors.	6-33
:CURSor[:TY]:HORizontal:POSition<x>	Sets the H cursor position or queries the current setting.	6-33
:CURSor[:TY]:HORizontal:TRACe	Sets the target waveform of the H cursor or queries the current setting.	6-33
:CURSor[:TY]:HORizontal:Y<x>?	Queries the Y-axis value of the H cursor.	6-33
:CURSor[:TY]:LOGic?	Queries all settings related to the logic display of the T-Y display.	6-33
:CURSor[:TY]:LOGic:FORMat	Sets the numerical notation of the logic display of the T-Y display.	6-33
:CURSor[:TY]:LOGic:ORDER	Sets the higher/lower bit of the logic display of the T-Y display.	6-33
:CURSor[:TY]:MARKer?	Queries all settings related to the marker cursor.	6-33
:CURSor[:TY]:MARKer:DX<x>?	Queries the X-axis value between the marker cursors.	6-34
:CURSor[:TY]:MARKer:DY<x>?	Queries the physical value of the Y-axis between the marker cursors.	6-34
:CURSor[:TY]:MARKer:M<x>?	Queries all settings related to the specified marker cursor.	6-34
:CURSor[:TY]:MARKer:M<x>:JUMP	Jumps the marker cursor onto the zoom waveform.	6-34
:CURSor[:TY]:MARKer:M<x>:POSition	Sets the X-axis value at the marker cursor position or queries the current setting.	6-34
:CURSor[:TY]:MARKer:M<x>:TRACe	Sets the target waveform of the marker cursor or queries the current setting.	6-34
:CURSor[:TY]:MARKer:X<x>?	Queries the X-axis value at the marker cursor position.	6-34
:CURSor[:TY]:MARKer:Y<x>?	Queries the Y-axis value at the marker cursor position.	6-34
:CURSor[:TY]:TYPE	Sets the cursor type or queries the current setting.	6-34
:CURSor[:TY]:VERTical?	Queries all settings related to V cursor.	6-34
:CURSor[:TY]:VERTical:DX?	Queries the X-axis value between the V cursors.	6-35
:CURSor[:TY]:VERTical:DY?	Queries the Y-axis value between the V cursors.	6-35
:CURSor[:TY]:VERTical:JUMP	Jumps the V cursor onto the zoom waveform.	6-35
:CURSor[:TY]:VERTical:PERDt?	Queries the 1/ s DT value between V cursors.	6-35
:CURSor[:TY]:VERTical:POSition<x>	Sets the V cursor position or queries the current setting.	6-35
:CURSor[:TY]:VERTical:TRACe	Sets the target waveform of the V cursor or queries the current setting.	6-35
:CURSor[:TY]:VERTical:X<x>?	Queries the X-axis value at the V cursor position.	6-35
:CURSor[:TY]:VERTical:Y<x>?	Queries the Y-axis value at the V cursor position.	6-35
:CURSor:XY?	Queries all settings related to the cursor of the X-Y display.	6-35
:CURSor:XY:HORizontal?	Queries all settings related to the H cursor of the X-Y display.	6-35
:CURSor:XY:HORizontal:DY?	Queries the Y-axis value between H cursors of the X-Y display.	6-36
:CURSor:XY:HORizontal:POSition<x>	Sets the H cursor position of the X-Y display or queries the current setting.	6-36
:CURSor:XY:HORizontal:TRACe	Sets the target waveform of the H cursor of the X-Y display when the X-axis setting is Quad or queries the current setting.	6-36
:CURSor:XY:HORizontal:STRACe	Sets the target waveform of the H cursor of the X-Y display when the X-axis setting is Single or queries the current setting.	6-36
:CURSor:XY:HORizontal:Y<x>?	Queries the Y-axis value of the H cursor of the X-Y display.	6-36
:CURSor:XY:MARKer?	Queries all settings related to the marker cursor of the X-Y display.	6-36
:CURSor:XY:MARKer:POSition	Sets the X-axis value at the marker cursor position of the X-Y display or queries the current setting.	6-36
:CURSor:XY:MARKer:T?	Queries the time value of the marker cursor of the X-Y display.	6-36
:CURSor:XY:MARKer:TRACe	Sets the target waveform of the marker cursor of the X-Y display when the X-axis setting is Quad or queries the current setting.	6-36
:CURSor:XY:MARKer:STRACe	Sets the target waveform of the marker cursor of the X-Y display when the X-axis setting is Single or queries the current setting.	6-36
:CURSor:XY:MARKer:X?	Queries the X-axis value at the marker cursor position of the X-Y display.	6-36
:CURSor:XY:MARKer:Y?	Queries the Y-axis value at the marker cursor position of the X-Y display.	6-36
:CURSor:XY:TYPE	Sets the cursor type of the X-Y display or queries the current setting.	6-37
:CURSor:XY:VERTical?	Queries all settings related to the V cursor of the X-Y display.	6-37
:CURSor:XY:VERTical:DX?	Queries the X-axis value between V cursors of the X-Y display.	6-37
:CURSor:XY:VERTical:POSition<x>	Sets the V cursor position of the X-Y display or queries the current setting.	6-37
:CURSor:XY:VERTical:TRACe	Sets the target waveform of the V cursor of the X-Y display when the X-axis setting is Quad or queries the current setting.	6-37
:CURSor:XY:VERTical:STRACe?	Queries the target waveform of the V cursor of the X-Y display when the X-axis setting is Single.	6-37
:CURSor:XY:VERTical:X<x>?	Queries the X-axis value at the V cursor position of the X-Y display.	6-37

## 6.1 A List of Commands

Command	Function	Page
<b>DISPlay Group</b>		
:DISPlay?	Queries all settings related to the display.	6-40
:DISPlay:ACCumulate?	Queries all settings related to the accumulated display of waveforms.	6-40
:DISPlay:Accumulate:MODE	Sets the accumulate mode or queries the current setting.	6-40
:DISPlay:ACCumulate:PERSistence	Sets the accumulate count or queries the current setting.	6-41
:DISPlay:COLor?	Queries all settings related to the color.	6-41
:DISPlay:COLor: {CHANnel<x>   MATH<x>   PodA   PodB   EVENT}	Sets the color of the waveform or queries the current setting.	6-41
:DISPlay:COLor: {BACK   CAPTURE}	Sets the color of the background or the sub waveform window or queries the current setting.	6-41
:DISPlay:COLor: BASecolor	Sets the basic color of the menu or queries the current setting.	6-41
:DISPlay:DECimation	Sets the number of decimation points or queries the current setting.	6-41
:DISPlay:EWINDOW	Turns ON/OFF the extra window or queries the current setting.	6-41
:DISPlay:FORMat	Sets the display format or queries the current setting.	6-41
:DISPlay:GRATICule	Sets the grid (graticule) or queries the current setting.	6-41
:DISPlay:INTENsity?	Queries all settings related to the brightness of the displayed items.	6-41
:DISPlay:INTENsity: {CURSor   MENU   GRID   SNAP   WAVEform   MARKer   CAPTURE   BACK}	Sets the brightness of the display item or queries the current setting.	6-41
:DISPlay:INTerpolate	Sets the interpolation method or queries the current setting.	6-42
:DISPlay:MONitor	Sets the monitor display or queries the current setting.	6-42
:DISPlay:MAPPING	Sets the mode of the waveform mapping to the split screen or queries the current setting.	6-42
:DISPlay:SMAPPING?	Queries all the settings related to the waveform mapping to the split screen.	6-42
:DISPlay:SMAPPING: {CHANnel<x>   MATH<x>   EVENT}	Sets the waveform mapping to the split screen or queries the current setting.	6-42
:DISPlay:SVALue	Turns ON/OFF the scale value display or queries the current setting.	6-42
:DISPlay:TLABEL	Turns ON/OFF the waveform label display or queries the current setting.	6-42
:DISPlay:TRANslucent	Turns ON/OFF the translucent mode of the pop-up menu or queries the current setting.	6-42
<b>EVENT Group</b>		
:EVENT?	Queries all settings related to the event waveform.	6-43
:EVENT:DISPlay	Turns ON/OFF the event waveform display or queries the current setting.	6-43
:EVENT:Label	Sets the event waveform label or queries the current setting.	6-43
:EVENT:VOICe*	Sets the voice event of the event waveform or queries the current setting.	6-43
:EVENT:DCAPuture	Sets the dual capture event of the event waveform or queries the current setting.	6-43
:EVENT:POSITION	Sets the vertical position of the event waveform or queries the current setting.	6-43
:EVENT:ZOOM	Sets the vertical zoom of the event waveform or queries the current setting.	6-43
* Cannot be used on the current DL750 that you are using.		
<b>FILE Group</b>		
:FILE?	Queries all settings related to the specified storage medium.	6-47
:FILE:DELeTe: {ASCIi   BINary   BMP   FLOat   MEASure   POSTscript   REALtime   SETup   SNAP   JPEG   PNG} [:EXECute]	Executes the deletion of the data file.	6-47
:FILE:DIRectory?	Queries all settings related to the current storage medium.	6-47
:FILE[:DIRectory]:CDIRectory	Changes the current directory.	6-47
:FILE[:DIRectory]:DRIVE	Sets the current storage medium.	6-47
:FILE[:DIRectory]:FILTer	Sets the filter or queries the current setting.	6-47
:FILE[:DIRectory]:FREE?	Queries the free disk space (bytes) on the current storage medium.	6-47
:FILE[:DIRectory]:MDIRectory	Creates a directory in the current directory.	6-47
:FILE[:DIRectory]:PATH?	Queries the current directory.	6-47
:FILE:LOAD: {BINary   REALtime   SETup   SNAP} :ABORT	Aborts the loading of the data.	6-47
:FILE:LOAD: {BINary   REALtime   SETup   SNAP} [:EXECute]	Executes the loading of the data.	6-47
:FILE:SAVE?	Queries all settings related to the saving of files.	6-47
:FILE:SAVE:ANAMing	Sets the auto naming function of the files to be saved or queries the current setting.	6-48
:FILE:SAVE: {ASCIi   BINary   FLOat} ?	Queries all settings related to the saving of the data.	6-48
:FILE:SAVE: {ASCIi   BINary   FLOat   MEASure   SETup   SNAP} :ABORT	Aborts the save operation.	6-48
:FILE:SAVE: {ASCIi   BINary   FLOat   MEASure   SETup   SNAP} [:EXECute]	Executes the saving of the file.	6-48
:FILE:SAVE: {ASCIi   BINary   FLOat} :HISTory	Sets the number of history pages to be saved or queries the current setting.	6-48

Command	Function	Page
:FILE:SAVE:{ASCIi BINary FLOat}:RANGe	Sets the range of data to be saved or queries the current setting.	6-48
:FILE:SAVE:{ASCIi BINary FLOat}:TRACe	Sets the waveform to be saved or queries the current setting.	6-48
:FILE:SAVE:ASCIi:INTerVal	Sets the decimation interval for saving data or queries the current setting.	6-48
:FILE:SAVE:BINary:COMPRESSion	Sets the data compression of the waveform to be saved or queries the current setting.	6-48
:FILE:SAVE:COMMENT	Sets the comment or queries the current setting.	6-48
:FILE:SAVE:NAME	Sets the name of the file to be saved or queries the current setting.	6-48
<b>GONogo Group</b>		
:GONogo?	Queries all settings related to the GO/NO-GO determination.	6-52
:GONogo:ABORT	Aborts the GO/NO-GO determination.	6-52
:GONogo:ACONdition	Sets the criteria for the action of the GO/NO-GO determination or queries the current setting.	6-52
:GONogo:ACTion?	Queries all settings related to the action taken when the execution condition is met and the criteria values.	6-52
:GONogo:ACTion:BUZZer	Sets whether an alarm is sounded when the execution condition is met or queries the current setting.	6-52
:GONogo:ACTion:HCOpy	Sets whether to output a hard copy to the built-in printer or an external printer when the execution condition is met or queries the current setting.	6-52
:GONogo:ACTion:IMAGe	Sets whether to save the screen image to the external storage medium when the execution condition is met or queries the current setting.	6-52
:GONogo:ACTion:MAIL?	Queries all settings related to the mail transmission when the execution condition is met.	6-53
:GONogo:ACTion:MAIL:COUNT	Sets the upper limit on the number of transmitted mail when the execution condition is met or queries the current setting.	6-53
:GONogo:ACTion:MAIL:MODE	Sets whether to send a mail when the execution condition is met or queries the current setting.	6-53
:GONogo:ACTion:SAVE	Sets whether to save the data to the external storage medium when the execution condition is met or queries the current setting.	6-53
:GONogo:COUNT?	Queries the number of GO/NO-GO determinations.	6-53
:GONogo:EXECute	Executes the GO/NO-GO determination.	6-53
:GONogo:LOGic	Sets the criteria logic of the GO/NO-GO determination or queries the current setting.	6-53
:GONogo:MODE	Sets the type of GO/NO-GO determination or queries the current setting.	6-53
:GONogo:NGCount?	Queries the number of NO-GOs of the GO/NO-GO determination.	6-53
:GONogo:PARAmeter?	Queries all settings related to parameter determination.	6-53
:GONogo:PARAmeter:ITEM<x>?	Queries all settings related to waveform parameter.	6-54
:GONogo:PARAmeter:ITEM<x>:CAUSE?	Queries whether the waveform parameter is the cause of NO-GO.	6-54
:GONogo:PARAmeter:ITEM<x>:MODE	Turns OFF the waveform parameter or sets the determination criteria of the waveform parameter or queries the current setting.	6-54
:GONogo:PARAmeter:ITEM<x>:TRACe	Sets the target waveform of the waveform parameter or queries the current setting.	6-54
:GONogo:PARAmeter:ITEM<x>:TYPE?	Queries the item and the upper and lower limits of the waveform parameter.	6-54
:GONogo:PARAmeter:ITEM<x>:TYPE:<parameter>	Sets the upper and lower limits of the measured item or queries the current setting.	6-54
:GONogo:PARAmeter:ITEM<x>:VALue?	Queries the measured value of the waveform parameter.	6-54
:GONogo:REMOte	Sets the remote mode of the GO/NO-GO determination or queries the current setting.	6-54
:GONogo:SEQuence	Sets the action count of the GO/NO-GO determination or queries the current setting.	6-54
:GONogo:STOPcount	Sets the waveform acquisition count of the GO/NO-GO determination or queries the current setting.	6-55
:GONogo:TRANge	Sets the determination range of GO/NO-GO determination or queries the current setting.	6-55
:GONogo:WAIT?	Waits for the completion of the GO/NO-GO determination.	6-55
:GONogo:ZONE?	Queries all settings related to the zone determination.	6-55
:GONogo:ZONE:EDIT<x>:EXIT	Exits from the edit menu of the zone of zone determination.	6-55
:GONogo:ZONE:EDIT<x>:PART	Edits a portion of the zone for zone determination.	6-55
:GONogo:ZONE:EDIT<x>:WHOLe	Edits the zone for zone determination.	6-56
:GONogo:ZONE:EDIT<x>:TRACe	Sets the target channel for the new zone waveform or queries the current setting.	6-56
:GONogo:ZONE:PATTern<x>?	Queries all settings related to the zone determination of the specified pattern.	6-56
:GONogo:ZONE:PATTern<x>:CAUSE?	Queries whether the pattern of the zone determination is the cause of NO-GO.	6-56

## 6.1 A List of Commands

Command	Function	Page
:GONogo:ZONE:PATtern<x>:MODE	Turns OFF the pattern of zone determination or sets the determination criteria or queries the current setting.	6-56
:GONogo:ZONE:PATtern<x>:SEND	Sets the zone data specified for the determination zone or queries the current setting.	6-56
:GONogo:ZONE:PATtern<x>:TRACe	Sets the target waveform for the pattern of the zone determination or queries the current setting.	6-56
:GONogo:ZONE:PATtern<x>:ZONE	Sets the target zone data of the pattern of the zone determination or queries the current setting.	6-56
<b>HCOPY Group</b>		
:HCOPY?	Queries all settings related to the output of screen data.	6-58
:HCOPY:ABORT	Aborts data output and paper feeding.	6-58
:HCOPY:COMMeNt	Sets the comment at the lower right corner of the screen or queries the current setting.	6-58
:HCOPY:DIReCtion	Sets the data output destination (print destination) or queries the current setting.	6-58
:HCOPY:EXECute	Executes the data output.	6-58
:HCOPY:EXTPrinter?	Queries all settings related to the external printer output.	6-58
:HCOPY:EXTPrinter:PRINter?	Queries the address of the printer that is currently connected.	6-59
:HCOPY:EXTPrinter:SElect	Sets the address of the printer or queries the current setting.	6-59
:HCOPY:EXTPrinter:TONE	Sets the half tone of the external printer output or queries the current setting.	6-59
:HCOPY:EXTPrinter:TYPE	Sets the type of output commands to send to the external printer or queries the current setting.	6-59
:HCOPY:NETPrint?	Queries all settings related to the network printer output.	6-59
:HCOPY:NETPrint:TONE	Sets the half tone of the network printer output or queries the current setting.	6-59
:HCOPY:NETPrint:TYPE	Sets the type of output commands to send to the network printer or queries the current setting.	6-59
:HCOPY:PRINter?	Queries all settings related to the built-in printer output.	6-59
:HCOPY:PRINter:FEED	Executes paper feeding of the built-in printer.	6-59
:HCOPY:PRINter:MAG	Sets the magnification for long copy or queries the current setting.	6-59
:HCOPY:PRINter:RANGE	Sets the output range for long copy or queries the current setting.	6-59
:HCOPY:PRINter:TDIV	Sets the T/div value for long copy or queries the current setting.	6-60
:HCOPY:PRINter:TYPE	Sets normal copy or long copy or queries the current setting.	6-60
<b>HISTORY Group</b>		
:HISTORY?	Queries all settings related to the history memory function.	6-62
:HISTORY:ABORT	Aborts the search.	6-62
:HISTORY:CLear	Executes history clear.	6-62
:HISTORY:DISPlay	Sets the start number or end number of the display record or queries the current setting.	6-62
:HISTORY:DMODE	Sets whether to display a single record or display all records accumulated.	6-62
:HISTORY:EXECute	Executes the search.	6-62
:HISTORY:RECORD	Sets the target record or queries the current setting.	6-62
:HISTORY:RECORD?MINimum	Queries the minimum record number.	6-62
:HISTORY:SMODE	Sets the history memory search mode or queries the current setting.	6-62
:HISTORY:TIME?	Queries the time of the target record number.	6-62
:HISTORY:ZONE?	Queries all settings related to the zone search.	6-62
:HISTORY:ZONE:EDIT<x>?	Queries all settings related to the specified zone cursor.	6-63
:HISTORY:ZONE:EDIT<x>:CONDition	Sets the search condition of the specified zone cursor or queries the current setting.	6-63
:HISTORY:ZONE:EDIT<x>:HORizontal	Sets the vertical position of the specified zone cursor or queries the current setting.	6-63
:HISTORY:ZONE:EDIT<x>:SOURce	Sets the target waveform of the specified zone cursor or queries the current setting.	6-63
:HISTORY:ZONE:EDIT<x>:VERTical	Sets the horizontal position of the specified zone cursor or queries the current setting.	6-63
:HISTORY:ZONE:LOGic	Sets the search logic condition between the specified zone cursors or queries the current setting.	6-63
<b>IMAGE Group</b>		
:IMAGE?	Queries all settings of the screen image data output.	6-65
:IMAGE:ABORT	Aborts saving the screen image data to the storage medium.	6-65
:IMAGE:COMMeNt	Sets the comment on the screen or queries the current setting.	6-65
:IMAGE:COMPReSSion	Sets the screen data compression of the BMP format or queries the current setting.	6-65
:IMAGE:EXECute	Saves the screen image data to the storage medium.	6-65
:IMAGE:FORMat	Sets the output format of the screen image data or queries the current setting.	6-65

Command	Function	Page
:IMAGE:SAVE?	Queries all settings related to the saving of the screen image data to the storage medium.	6-65
:IMAGE:SAVE:ANAMing	Sets the auto naming function of file names or queries the current setting.	6-65
:IMAGE:SAVE:CDIRectory	Changes the directory in which to create the file.	6-65
:IMAGE:SAVE:DRIVE	Sets the storage medium on which to create the file or queries the current setting.	6-65
:IMAGE:SAVE:NAME	Sets the name of the file to be created or queries the current setting.	6-65
:IMAGE:SAVE:PATH?	Queries the current directory.	6-65
:IMAGE:SEND?	Queries the screen image data.	6-65
:IMAGE:TONE	Sets the color tone of the screen image data to be output or queries the current setting.	6-65
<b>INITialize Group</b>		
:INITialize:EXECute	Executes the initialization.	6-66
:INITialize:UNDO	Cancel the execution of the initialization.	6-66
<b>LOGic Group</b>		
:LOGic?	Queries all settings related to the logic waveforms.	6-68
:LOGic:{PODA PODB}?	Queries all settings related to the specified logic waveform.	6-68
:LOGic:{PODA PODB}:ALL:DISPlay	Turns ON/OFF the display of all bits of the specified logic waveform.	6-68
:LOGic:{PODA PODB}:{BIT1~8}:DISPlay	Sets the display of the bit of the specified logic waveform or queries the current setting.	6-68
:LOGic:{PODA PODB}:{BIT1~8}:LAbel	Sets the label of the bit of the specified logic waveform or queries the current setting.	6-68
:LOGic:{PODA PODB}:BMApping	Turns ON/OFF the bit display position of the specified logic waveform or queries the current setting.	6-68
:LOGic:{PODA PODB}:DISPlay	Turns ON/OFF the display of all bits of the specified logic waveform or queries the current setting.	6-68
:LOGic:{PODA PODB}:LAbel	Sets the label of the specified logic waveform or queries the current setting.	6-68
:LOGic:{PODA PODB}:POStion	Sets the vertical position of the specified logic waveform or queries the current setting.	6-69
:LOGic:{PODA PODB}:ZOOM	Sets the vertical zoom rate of the specified logic waveform or queries the current setting.	6-69
<b>LStart Group</b>		
:LStart	Executes log start.	6-69
:LStart?	Waits for the completion of the log start execution.	6-69
<b>MATH Group</b>		
:MATH<x>?	Queries all settings related to the computation.	6-72
:MATH<x>:AVERage?	Queries all settings related to averaging of the user-defined computation.	6-72
:MATH<x>:AVERage:CCount	Sets the cycle count of the cycle average or queries the current setting.	6-72
:MATH<x>:AVERage:COUNT	Sets the waveform acquisition count of linear average or queries the current setting.	6-73
:MATH<x>:AVERage:EWEight	Sets the attenuation constant of exponential averaging or queries the current setting.	6-73
:MATH<x>:AVERage:MODE	Sets the averaging mode or queries the current setting.	6-73
:MATH<x>:AVERage:TYPE	Sets the averaging area or queries the current setting.	6-73
:MATH<x>:BINary?	Queries all settings related to the binary computation.	6-73
:MATH<x>:BINary:{CHANnel<x> MATH<x>}	Sets the threshold level of the binary computation channel or queries the current setting.	6-73
:MATH<x>:CONStant<x>	Sets the constant of the user-defined computation or queries the current setting.	6-73
:MATH<x>:DEFine	Sets the equation of the user-defined computation or queries the current setting.	6-73
:MATH<x>:DISPlay	Sets whether to display computed waveforms or queries the current setting.	6-73
:MATH<x>:EXECute	Executes the computation.	6-73
:MATH<x>:FILTer<x>?	Queries all settings related to the filter of the user-defined computation.	6-74
:MATH<x>:FILTer<x>:BAND	Sets the filter bandwidth or queries the current setting.	6-74
:MATH<x>:FILTer<x>:CUTOff<x>	Sets the cutoff frequency or queries the current setting.	6-74
:MATH<x>:FILTer<x>:TYPE	Sets the filter type or queries the current setting.	6-74
:MATH<x>:FFT?	Queries all settings related to the FFT computation.	6-74
:MATH<x>:FFT:POINt	Sets the number of points of the FFT computation or queries the current setting.	6-74
:MATH<x>:FFT:WINDow	Sets the window function of the FFT computation or queries the current setting.	6-74



## 6.1 A List of Commands

Command	Function	Page
:MATH<x>:LABEL	Sets the label of the computation or queries the current setting.	6-74
:MATH<x>:MODE	Turns ON/OFF the computation or queries the current setting.	6-74
:MATH<x>:MREFERENCE	Sets the computation range or queries the current setting.	6-74
:MATH<x>:OPERATION	Sets the computation details or queries the current setting.	6-75
:MATH<x>:SCALE?	Queries all settings related to scaling.	6-75
:MATH<x>:SCALE:MODE	Sets the scaling mode or queries the current setting.	6-75
:MATH<x>:SCALE:VALUE	Sets the upper and lower limits for manual scaling or queries the current setting.	6-75
:MATH<x>:SHIFT { CHANNEL <x>   MATH <x> }	Sets the number of shift points or queries the current setting.	6-75
:MATH<x>:UNIT	Sets the unit to be added to the computed result or queries the current setting.	6-75
<b>MEASURE Group</b>		
:MEASURE?	Queries all settings related to the automated measurement of waveform parameters.	6-78
:MEASURE: { CHANNEL <x>   MATH <x> } ?	Turns ON/OFF all the parameters of the channel or queries the current setting.	6-85
:MEASURE: { CHANNEL <x>   MATH <x> } : ALL	Turns ON/OFF collectively all the measurement items of the channel.	6-85
:MEASURE: { CHANNEL <x>   MATH <x> } : COPY	Copies the ON/OFF setting of all measurement items of a specified channel to all other channels.	6-85
:MEASURE: { CHANNEL <x>   MATH <x> } : DELAY?	Queries all settings related to the delay between channels of the waveform.	6-85
:MEASURE: { CHANNEL <x>   MATH <x> } : DELAY: MEASURE?	Queries all settings of the source waveform of the delay between channels.	6-85
:MEASURE: { CHANNEL <x>   MATH <x> } : DELAY: MEASURE: COUNT	Sets the number of counts of the source waveform of the delay between channels.	6-85
:MEASURE: { CHANNEL <x>   MATH <x> } : DELAY: MEASURE: SLOPE	Sets the slope of the source waveform of the delay between channels.	6-86
:MEASURE: { CHANNEL <x>   MATH <x> } : DELAY: REFERENCE?	Queries all settings of the reference waveform of the delay between channels.	6-86
:MEASURE: { CHANNEL <x>   MATH <x> } : DELAY: REFERENCE: COUNT	Sets the number of counts of the reference waveform of the delay between channels.	6-86
:MEASURE: { CHANNEL <x>   MATH <x> } : DELAY: REFERENCE: SLOPE	Sets the slope of the reference waveform of the delay between channels.	6-86
:MEASURE: { CHANNEL <x>   MATH <x> } : DELAY: REFERENCE: SOURCE	Sets the reference of the delay between channels to the trigger point or the waveform or queries the current setting.	6-86
:MEASURE: { CHANNEL <x>   MATH <x> } : DELAY: REFERENCE: TRACE	Sets the trace of the reference waveform of the delay between channels.	6-86
:MEASURE: { CHANNEL <x>   MATH <x> } : DELAY: STATE	Sets the display format of the delay parameter or queries the current setting.	6-86
:MEASURE: { CHANNEL <x>   MATH <x> } : DELAY: VALUE?	Queries the value of the delay parameter of the waveform that is measured automatically.	6-86
:MEASURE: { CHANNEL <x>   MATH <x> } : DPRXIMAL?	Queries all settings related to distal, mesial, and proximal.	6-87
:MEASURE: { CHANNEL <x>   MATH <x> } : DPRXIMAL: MODE	Sets the mode of the distal, mesial, and proximal points or queries the current setting.	6-87
:MEASURE: { CHANNEL <x>   MATH <x> } : DPRXIMAL: PERCENT	Sets the distal, mesial, and proximal points or queries the current setting (%).	6-87
:MEASURE: { CHANNEL <x>   MATH <x> } : DPRXIMAL: UNIT	Sets the distal, mesial, and proximal points or queries the current setting (voltage).	6-87
:MEASURE: { CHANNEL <x>   MATH <x> } : METHOD	Sets the High and Low points or queries the current setting.	6-87
:MEASURE: { CHANNEL <x>   MATH <x> } : <parameter>?	Queries all settings related to the parameter of the waveform.	6-87
:MEASURE: { CHANNEL <x>   MATH <x> } : <parameter>: STATE	Turns ON/OFF the parameter of the waveform or queries the current setting.	6-88
:MEASURE: { CHANNEL <x>   MATH <x> } : <parameter>: VALUE?	Queries the value of the parameter of the waveform that is measured automatically.	6-88

Command	Function	Page
:MEASure:CYCLEmode	Sets the 1 cycle mode of the automated measurement of waveform parameters or queries the current setting.	6-88
:MEASure:MODE	Turns ON/OFF the automated measurement or queries the current setting.	6-88
:MEASure:TRANge	Sets the measurement range or queries the current setting.	6-88
:MEASure:WAIT?	Waits for the completion of the automated measurement.	6-88
<b>SEARCh Group</b>		
:SEARCh?	Queries all settings related to the search function.	6-91
:SEARCh:ASCRoll?	Queries all settings related to the auto scroll.	6-91
:SEARCh:ASCRoll:DIRection	Sets the direction for moving the zoom box of the auto scroll or queries the current setting.	6-91
:SEARCh:ASCRoll:SPeEd	Sets the speed for moving the zoom box of the auto scroll or queries the current setting.	6-91
:SEARCh:EDGE?	Queries all settings related to the edge search.	6-91
:SEARCh:EDGE:COUNT	Sets the edge search count or queries the current setting.	6-91
:SEARCh:EDGE:HYSteresis	Sets the hysteresis of the edge search level or queries the current setting.	6-91
:SEARCh:EDGE:LEVEl	Sets the edge search level or queries the current setting.	6-91
:SEARCh:EDGE:LOGic?	Queries all settings related to the logic channel of the edge search.	6-91
:SEARCh:EDGE:LOGic:{A1 A2 A3 A4 A5 A6 A7 A8 B1 B2 B3 B4 B5 B6 B7 B8}	Sets the slope of the bit of the logic channel of the edge search or queries the current setting.	6-91
:SEARCh:EDGE:LOGic:CONDition	Sets the condition of the bit of the logic channel of the edge search or queries the current setting.	6-91
:SEARCh:EDGE:SLOPe	Sets the edge polarity or queries the current setting.	6-92
:SEARCh:EDGE:SOURce	Sets the target waveform of the edge search or queries the current setting.	6-92
:SEARCh:EXECute	Executes the search.	6-92
:SEARCh:MAG<x>	Sets the zoom rate of the zoom window or queries the current setting.	6-92
:SEARCh:MLINKage	Turns ON/OFF the link when changing the zoom rate or queries the current setting.	6-92
:SEARCh:PLINKage	Turns ON/OFF the link when moving the zone box or queries the current setting.	6-92
:SEARCh:POSition<x>	Sets the zoom box position or queries the current setting.	6-92
:SEARCh:SELEct	Sets the storage number of the searched position or queries the position stored at the storage number.	6-92
:SEARCh:SELEct? MAXimum	Queries the maximum storage number of searched positions.	6-92
:SEARCh:SPOint	Sets the search start point or queries the current setting.	6-92
:SEARCh:TDIV<x>	Sets the T/div value of the zoom waveform or queries the current setting.	6-92
:SEARCh:TWINdow	Sets the zoom window on which the searched section is to be displayed or queries the current setting.	6-92
:SEARCh:TYPE	Sets the search type or queries the current setting.	6-92
<b>SNAP Group</b>		
:SNAP	Executes the snapshot.	6-93
<b>SStArt Group</b>		
:SStArt	Executes single start.	6-93
:SStArt?	Waits for the completion of the single start execution.	6-93
<b>StARt Group</b>		
:StARt	Starts the waveform acquisition.	6-93
<b>StATus Group</b>		
:StATus?	Queries all settings related to the communication status function.	6-94
:StATus:CONDition?	Queries the condition register.	6-94
:StATus:EESE	Sets the extended event enable register or queries the current setting.	6-94
:StATus:EESR?	Queries the extended event register and clears the register.	6-94
:StATus:ERRor?	Queries the code and description of the error that occurred.	6-94
:StATus:FILTer<x>	Sets the transition filter or queries the current setting.	6-95
:StATus:QENable	Sets whether or not to store messages other than errors to the error queue or queries the current setting.	6-95
:StATus:QMESsage	Sets whether to attach a message to the response to the :StATus:ERRor? query or queries the current setting.	6-95
:StATus:SPOLL?	Executes the serial polling.	6-95
<b>StOP Group</b>		
:StOP	Stops the waveform acquisition.	6-95

## 6.1 A List of Commands

Command	Function	Page
<b>SYSTEM Group</b>		
:SYSTem?	Queries all settings related to the system.	6-97
:SYSTem:CLICk	Turns ON/OFF the click sound or queries the current setting.	6-97
:SYSTem:CLOCk?	Queries all settings related to the date and time.	6-97
:SYSTem:CLOCk:DATE	Sets the date or queries the current setting.	6-97
:SYSTem:CLOCk:MODE	Turns ON/OFF the date/time display or queries the current setting.	6-97
:SYSTem:CLOCk:TIME	Sets the time or queries the current setting.	6-97
:SYSTem:HDMotor	Turns ON/OFF the motor of the internal hard disk or queries the current setting.	6-98
:SYSTem:LANguage	Sets the message language or queries the current setting.	6-98
:SYSTem:LCD?	Queries all settings related to the LCD.	6-98
:SYSTem:LCD:AUTO?	Queries all settings related to the auto off function of the LCD backlight.	6-98
:SYSTem:LCD:AUTO:MODE	Turns ON/OFF the auto off function of the LCD backlight or queries the current setting.	6-98
:SYSTem:LCD:AUTO:TIME	Sets the auto off time of the LCD backlight or queries the current setting.	6-98
:SYSTem:LCD:BRIGHtness	Sets the brightness of the LCD or queries the current setting.	6-98
:SYSTem:LCD:MODE	Turns ON/OFF the LCD backlight or queries the current setting.	6-98
:SYSTem:MLANguage	Sets the menu language or queries the current setting.	6-98
:SYSTem:OVERview	Displays system information.	6-98
:SYSTem:PACTion	Sets whether to enable the action-on-trigger mode at power ON or queries the current setting.	6-98
:SYSTem:PStart	Sets whether to start the waveform acquisition at power ON or queries the current setting.	6-98
:SYSTem:SCSI?	Queries all settings related to the SCSI-ID.	6-98
:SYSTem:SCSI:INITialize	Executes the initialization of SCSI related parameters.	6-98
:SYSTem:SCSI:INTernalId	Set the SCSI-ID of the internal hard disk or queries the current settings.	6-98
:SYSTem:SCSI:OWNId	Set the SCSI-ID of the DL750 or queries the current settings.	6-99
:SYSTem:VIDeo	Turns ON/OFF the video output or queries the current setting.	6-99
<b>TIMEbase Group</b>		
:TIMEbase?	Queries all settings related to the time base.	6-100
:TIMEbase:SOURce	Sets the time base or queries the current setting.	6-100
:TIMEbase:SRATE?	Queries the sample rate or queries the current setting.	6-100
:TIMEbase:TDIV	Sets the T/div value or queries the current setting.	6-100
<b>TRIGger Group</b>		
:TRIGger?	Queries all settings related to the trigger.	6-107
:TRIGger:ABN?	Queries all settings related to the A->B(n) trigger.	6-108
:TRIGger:ABN:COUNT	Sets the number times condition B is to be satisfied or queries the current setting.	6-108
:TRIGger:ACTion?	Queries all settings related to the action-on-trigger.	6-108
:TRIGger:ACTion:BUZZer	Sets whether a beep is sounded when a action is activated or queries the current setting.	6-108
:TRIGger:ACTion:HCOpy	Sets whether to output a hard copy to the built-in printer or an external printer when a action is activated or queries the current setting.	6-108
:TRIGger:ACTion:IMAGe	Sets whether to save the screen image to the storage medium when an action is activated or queries the current setting.	6-108
:TRIGger:ACTion:MAIL?	Queries all settings related to the mail transmission when an action is activated.	6-108
:TRIGger:ACTion:MAIL:COUNT	Sets the upper limit on the number of transmitted mail when an action is activated or queries the current setting.	6-108
:TRIGger:ACTion:MAIL:MODE	Queries whether a mail is set to be transmitted when an action is activated.	6-108
:TRIGger:ACTion:MODE	Sets the action-on-trigger mode or queries the current setting.	6-108
:TRIGger:ACTion:SAVE	Sets whether to save the waveform data to the storage medium when an action is activated or queries the current setting.	6-109
:TRIGger:ACTion:SEQuence	Sets whether to repeat the action or queries the current setting.	6-109
:TRIGger:ADB?	Queries all settings related to the A Delay B(n) trigger.	6-109
:TRIGger:ADB:DELay	Sets the delay time of condition B or queries the current setting.	6-109
:TRIGger:ATRigger?	Queries all settings related to condition A.	6-109
:TRIGger:ATRigger:CHANnel<x>	Sets the channel condition of condition A or queries the current setting.	6-109
:TRIGger:ATRigger:LOGic?	Queries all settings related the logic of condition A.	6-109
:TRIGger:ATRigger:LOGic:{A1~B8}	Sets the logic bit of condition A or queries the current setting.	6-109
:TRIGger:ATRigger:{PODA PODB}	Sets the logic condition of condition A or queries the current setting.	6-109
:TRIGger:ATRigger:CONDition	Sets the condition to be satisfied for condition A or queries the current setting.	6-109
:TRIGger:BGTime?	Queries all settings related to the B>Time trigger.	6-110
:TRIGger:BGTime:TIME	Sets the pulse width of the B>Time trigger or queries the current setting.	6-110

Command	Function	Page
:TRIGger:BLTime?	Queries all settings related to the B<Time trigger.	6-110
:TRIGger:BLTime:TIME	Sets the pulse width of the B<Time trigger or queries the current setting.	6-110
:TRIGger:BTOut?	Queries all settings related to the B Time Out trigger.	6-110
:TRIGger:BTOut:TIME	Sets the pulse width of the B Time Out trigger or queries the current setting.	6-110
:TRIGger:BTRigger?	Queries all settings related to condition B.	6-110
:TRIGger:BTRigger:CHANnel<x>	Sets the channel condition of condition B or queries the current setting.	6-110
:TRIGger:BTRigger:LOGic?	Queries all settings related the logic of condition B.	6-110
:TRIGger:BTRigger:LOGic:{A1~B8}	Sets the logic bit of condition B or queries the current setting.	6-110
:TRIGger:BTRigger:{PODA PODB}	Sets the logic condition of condition B or queries the current setting.	6-111
:TRIGger:BTRigger:CONDition	Sets the condition to be satisfied for condition B or queries the current setting.	6-111
:TRIGger:DElay	Sets the delay or queries the current setting.	6-111
:TRIGger:EOA?	Queries all settings related to the Edge on A trigger.	6-111
:TRIGger:EOA:CHANnel<x>	Sets the channel condition of the Edge on A trigger or queries the current setting.	6-111
:TRIGger:EOA:{PODA PODB}	Sets the logic condition of the Edge on A trigger or queries the current setting.	6-111
:TRIGger:EOA:CONDition	Sets the condition to be satisfied for the Edge on A trigger or queries the current setting.	6-111
:TRIGger:HOLDoff?	Queries all settings related to the hold off.	6-111
:TRIGger:HOLDoff:TIME	Sets the hold off time or queries the current setting.	6-111
:TRIGger:MODE	Sets the trigger mode or queries the current setting.	6-111
:TRIGger:OR?	Queries all settings related to the OR trigger.	6-111
:TRIGger:OR:CHANnel<x>	Sets the channel condition of the OR trigger or queries the current setting.	6-112
:TRIGger:OR:{PODA PODB}	Sets the logic condition of the OR trigger or queries the current setting.	6-112
:TRIGger:POSition	Sets the trigger position or queries the current setting.	6-112
:TRIGger:SCount	Sets the number of times the trigger is to be activated when the trigger mode is Single(N) or queries the current setting.	6-112
:TRIGger[:SIMple]?	Queries all settings related to the simple trigger.	6-112
:TRIGger[:SIMple]:HYSTeresis	Sets the hysteresis of the search level or queries the current setting.	6-112
:TRIGger[:SIMple]:LEVel	Sets the trigger level of the channel or queries the current setting.	6-112
:TRIGger[:SIMple]:SLOPe	Sets the trigger slope of the channel or queries the current setting.	6-112
:TRIGger:SIMple:SOURce	Sets the trigger source or queries the current setting.	6-112
:TRIGger[:SIMple]:LOGic?	Queries all settings related to the logic of the simple trigger.	6-112
:TRIGger[:SIMple]:LOGic:BIT	Sets the bit for setting the trigger slope or queries the current setting.	6-112
:TRIGger[:SIMple]:LOGic:SLOPe	Sets the logic trigger slope or queries the current setting.	6-113
:TRIGger:SOURce?	Queries all settings related to the trigger source of the enhanced trigger.	6-113
:TRIGger:SOURce:CHANnel<x>?	Queries all settings related to the trigger source of the specified channel.	6-113
:TRIGger:SOURce:CHANnel<x>:CENTer	Sets the center of the window trigger or queries the current setting.	6-113
:TRIGger:SOURce:CHANnel<x>:HYSTeresis	Sets the hysteresis of the search level or queries the current setting.	6-113
:TRIGger:SOURce:CHANnel<x>:LEVel	Sets the trigger level of the specified channel or queries the current setting.	6-113
:TRIGger:SOURce:CHANnel<x>:WIDTh	Sets the width of the window trigger or queries the current setting.	6-114
:TRIGger:SOURce:LOGic:{A1~A8 B1~B8}	Sets the logic bit condition or queries the current setting.	6-114
:TRIGger:TGTime?	Queries all settings related to the T>Time trigger.	6-114
:TRIGger:TGTime:TIME	Sets the pulse width of the T>Time trigger or queries the current setting.	6-114
:TRIGger:TImer?	Queries all settings related to the timer trigger.	6-114
:TRIGger:TImer:DATE	Sets the date of the timer trigger or queries the current setting.	6-114
:TRIGger:TImer:INTerval	Sets the trigger time interval of the timer trigger or queries the current setting.	6-114
:TRIGger:TImer:TIME	Sets the trigger time of the timer trigger or queries the current setting.	6-114
:TRIGger:TITime?	Queries all settings related to the TinTime trigger.	6-114
:TRIGger:TITime:TIME<x>	Sets the pulse width of the TinTime trigger or queries the current setting.	6-114
:TRIGger:TlTime?	Queries all settings related to the T<Time trigger.	6-114
:TRIGger:TlTime:TIME	Sets the pulse width of the T<Time trigger or queries the current setting.	6-114
:TRIGger:TOutTime?	Queries all settings related to the ToutTime trigger.	6-114
:TRIGger:TOutTime:TIME<x>	Sets the pulse width of the ToutTime trigger or queries the current setting.	6-115
:TRIGger:TYpe	Sets the trigger type or queries the current setting.	6-115
:TRIGger:WINDow?	Queries all settings related to the window trigger.	6-115
:TRIGger:WINDow:CHANnel<x>	Sets the channel condition of the window trigger or queries the current setting.	6-115
:TRIGger:WINDow:{PODA PODB}	Sets the logic condition of the window trigger or queries the current setting.	6-115
<b>WAVeform Group</b>		
:WAVeform?	Queries all information about the target waveform data.	6-117
:WAVeform:BITS?	Queries the bit length of the target waveform data.	6-117
:WAVeform:BYTeorder	Sets the transmission order of word format data or queries the current setting.	6-117
:WAVeform:CAPture?	Queries all information about the sub waveform.	6-117
:WAVeform:CAPture:END	Sets the last data point of the sub waveform or queries the current setting.	6-117

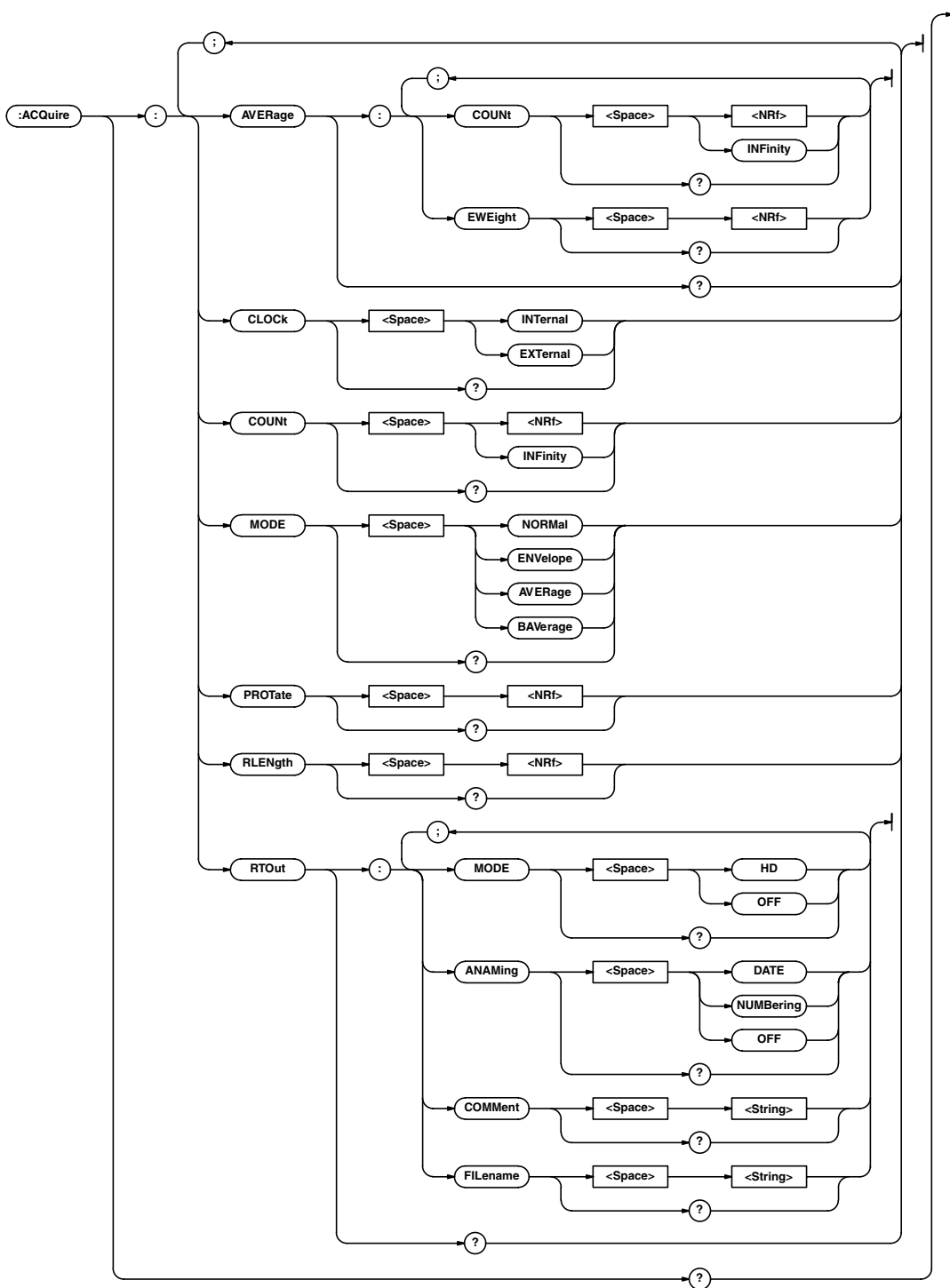
## 6.1 A List of Commands

Command	Function	Page
:WAVeform:CAPTure:LENGth?	Queries the total number of data points of the sub waveform.	6-117
:WAVeform:CAPTure:RECOrd	Sets the target record number of the sub waveform or queries the current setting.	6-117
:WAVeform:CAPTure:RECOrd?MAXimum	Queries the maximum record number.	6-117
:WAVeform:CAPTure:RECOrd?MINimum	Queries the minimum record number.	6-117
:WAVeform:CAPTure:SEND?	Queries the waveform data of the sub waveform.	6-117
:WAVeform:CAPTure:SRATe?	Queries the sample rate of the sub waveform.	6-118
:WAVeform:CAPTure:STARt	Sets the first data point of the sub waveform or queries the current setting.	6-118
:WAVeform:CAPTure:TRIGger?	Queries the trigger position of the sub waveform.	6-118
:WAVeform:DATaselect	Sets the data format used to query the waveform data or queries the current setting.	6-118
:WAVeform:END	Sets the end data point of the target waveform or queries the current setting.	6-118
:WAVeform:FORMat	Sets the format of the transmitted data or queries the current setting.	6-118
:WAVeform:LENGth?	Queries the total number of data points of the target waveform.	6-118
:WAVeform:MODule?	Queries the module of the target waveform.	6-118
:WAVeform:OFFSet?	Queries the offset value of the target waveform.	6-118
:WAVeform:RANGe?	Queries the range value of the target waveform.	6-118
:WAVeform:RECOrd	Sets the target record number or queries the current setting.	6-118
:WAVeform:RECOrd?MINimum	Queries the minimum record number.	6-119
:WAVeform:SEND?	Queries the target waveform data.	6-119
:WAVeform:SIGN?	Queries the existence of a sign of the target waveform data.	6-119
:WAVeform:SRATe?	Sets the sample rate of the target record or queries the current setting.	6-119
:WAVeform:STARt	Sets the start data point of the target waveform or queries the current setting.	6-119
:WAVeform:TRACe	Sets the target waveform or queries the current setting.	6-119
:WAVeform:TRIGger?	Sets the trigger position of the target record or queries the current setting.	6-119
:WAVeform:TYPE?	Queries the acquisition mode of the target waveform.	6-120
<b>XY Group</b>		
:XY<x>?	Queries all settings related to the X-Y display.	6-120
:XY<x>:MODE	Sets the display mode or queries the current setting.	6-120
:XY<x>:TRANge	Sets the range of the T-Y waveform to be displayed on the X-Y display or queries the current setting.	6-120
:XY<x>:XAXis	Sets whether to make the channel assigned to the X-axis common or queries the current setting.	6-121
:XY<x>:XTRace	Sets the channel assigned to the X-axis or queries the current setting.	6-121
:XY<x>:YTRace	Sets the channel assigned to the Y-axis or queries the current setting.	6-121
<b>ZOOM Group</b>		
:ZOOM?	Queries all settings related to the waveform zoom.	6-123
:ZOOM:ALLocation?	Queries all settings related to the zoom target waveform.	6-123
:ZOOM:ALLocation:ALL	Sets all waveforms to be zoomed.	6-123
:ZOOM:ALLocation:{CHANnel<x> MATH<x> PODA PODB}	Sets the zoom target waveform or queries the current setting.	6-124
:ZOOM:FITMeasure	Moves the range of the automated measurement of waveform parameters to the ends of the zoom box.	6-124
:ZOOM:MOVE	Moves the zoom box to the beginning of the waveform.	6-124
:ZOOM:FORMat	Sets the zoom waveform display format or queries the current setting.	6-124
:ZOOM:MLINKage	Turns ON/OFF the magnification link or queries the current setting.	6-124
:ZOOM:MAG<x>	Sets the zoom rate of the target zoom waveform window or queries the current setting.	6-124
:ZOOM:MODE	Sets the zoom waveform display mode or queries the current setting.	6-124
:ZOOM:PLINKage	Turns ON/OFF the link when moving the zone box or queries the current setting.	6-124
:ZOOM:POSition<x>	Sets the target zoom box position or queries the current setting.	6-124
:ZOOM:TDIV<x>	Sets the T/div value of the zoom waveform or queries the current setting.	6-124
<b>Common Command Group</b>		
*CAL?	Performs calibration and queries the result.	6-125
*CLS	Clears the standard event register, extended event register, and error queue.	6-125
*ESE	Sets the standard event enable register or queries the current setting.	6-125
*ESR?	Queries the standard event register and clears the register.	6-126
*IDN?	Queries the instrument model.	6-126
*LRN?	Queries collectively the current settings of the command group.	6-126
*OPC	Sets the bit 0 (OPC bit) of the standard event register to 1 upon the completion of the specified overlap command.	6-127
*OPC?	Creates a response after the completion of the specified overlap command.	6-127
*OPT?	Queries the options.	6-128

Command	Function	Page
*PSC	Sets whether or not to clear the registers at power on or queries the current setting.	6-128
*RST	Initializes the command group settings.	6-128
*SRE	Sets the service request enable register or queries the current setting.	6-128
*STB?	Queries the status byte register.	6-128
*WAI	Holds the subsequent command until the completion of the specified overlap operation.	6-128

### 6.2 ACQuire Group

The commands in this group deal with the waveform acquisition. You can make the same settings and inquiries as when the ACQ key on the front panel is used.



**:ACQUIRE?**

Function Queries all settings related to the waveform acquisition.

Syntax :ACQUIRE?

Example :ACQUIRE? -> :ACQUIRE:  
RLENGTH 10000;MODE NORMAL;  
PROTATE 1;AVERAGE:COUNT INFINITY;  
EWEIGHT 16;:ACQUIRE:COUNT INFINITY;  
CLOCK INTERNAL;RTOUT:MODE OFF;  
ANAMING DATE;FILENAME "";COMMENT ""

**:ACQUIRE:AVERAGE?**

Function Queries all settings related to averaging.

Syntax :ACQUIRE:AVERAGE?

Example :ACQUIRE:AVERAGE? -> :ACQUIRE:  
AVERAGE:COUNT INFINITY;EWEIGHT 16

**:ACQUIRE:AVERAGE:COUNT**

Function Sets the waveform acquisition count when in average mode or queries the current setting.

Syntax :ACQUIRE:AVERAGE:COUNT  
{<NRf>|INFINITY}  
:ACQUIRE:AVERAGE:COUNT?  
<NRf>=2 to 65536 (2<sup>n</sup> steps)

Example :ACQUIRE:AVERAGE:COUNT INFINITY  
:ACQUIRE:AVERAGE:COUNT? ->  
:ACQUIRE:AVERAGE:COUNT INFINITY

**:ACQUIRE:AVERAGE:EWEIGHT  
(Exponent WEIGHT)**

Function Sets the attenuation constant when averaging infinitely or queries the current setting.

Syntax :ACQUIRE:AVERAGE:EWEIGHT {<NRf>}  
:ACQUIRE:AVERAGE:EWEIGHT?  
<NRf>=2 to 256 (2<sup>n</sup> steps)

Example :ACQUIRE:AVERAGE:EWEIGHT 16  
:ACQUIRE:AVERAGE:EWEIGHT? ->  
:ACQUIRE:AVERAGE:EWEIGHT 16

**:ACQUIRE:CLOCK**

Function Sets the time base or queries the current setting.

Syntax :ACQUIRE:{INTERNAL|EXTERNAL}  
:ACQUIRE:CLOCK?

Example :ACQUIRE:CLOCK INTERNAL  
:ACQUIRE:CLOCK? -> :ACQUIRE:  
CLOCK INTERNAL

**:ACQUIRE:COUNT**

Function Sets the waveform acquisition count when in normal mode or queries the current setting.

Syntax :ACQUIRE:COUNT {<NRf>|INFINITY}  
:ACQUIRE:COUNT?  
<NRf>=2 to 65535

Example :ACQUIRE:COUNT INFINITY  
:ACQUIRE:COUNT? -> :ACQUIRE:  
COUNT INFINITY

**:ACQUIRE:MODE**

Function Sets the waveform acquisition mode or queries the current setting.

Syntax :ACQUIRE:MODE  
{AVERAGE|BAVERAGE|ENVELOPE|NORMAL}  
:ACQUIRE:MODE?

Example :ACQUIRE:MODE NORMAL  
:ACQUIRE:MODE? -> :ACQUIRE:  
MODE NORMAL

**:ACQUIRE:RLENGTH**

Function Sets the record length or queries the current setting.

Syntax :ACQUIRE:RLENGTH {<NRf>}  
:ACQUIRE:RLENGTH?  
<NRf>=1000 to 1000000000 (for the resolution, see the *User's Manual IM701210-01E*.)

Example :ACQUIRE:RLENGTH 1000  
:ACQUIRE:RLENGTH? -> :ACQUIRE:  
RLENGTH 1000

**:ACQUIRE:PROTATE**

Function Sets pulse/rotate when using external clock or queries the current setting.

Syntax :ACQUIRE:PROTATE {<NRf>}  
:ACQUIRE:PROTATE?  
<NRf>=1 to 24000

Example :ACQUIRE:PROTATE 100  
:ACQUIRE:PROTATE? -> :ACQUIRE:  
PROTATE 100

**:ACQUIRE:RTOUT?**

Function Queries all settings related to realtime recording.

Syntax :ACQUIRE:RTOUT?  
:ACQUIRE:RTOUT?

Example :ACQUIRE:RTOUT? -> :ACQUIRE:RTOUT:  
MODE OFF;ANAMING DATE;FILENAME "";  
COMMENT ""

Description

**:ACQUIRE:RTOUT:MODE**

Function Sets whether to perform realtime recording or queries the current setting.

Syntax :ACQUIRE:RTOUT:MODE {HD|OFF}

Example :ACQUIRE:RTOUT:MODE HD  
:ACQUIRE:RTOUT:MODE? -> :ACQUIRE:  
RTOUT:MODE HD

Description The internal hard disk is optional. If you select HD when there is no internal hard disk, an error occurs.

**:ACQUIRE:RTOUT:ANAMING**

Function Sets auto naming for realtime recording or queries the current setting.

Syntax :ACQUIRE:RTOUT:ANAMING  
{DATE|NUMBERING|OFF}

Example :ACQUIRE:RTOUT:ANAMING NUMBERING  
:ACQUIRE:RTOUT:ANAMING? ->  
:ACQUIRE:RTOUT:ANAMING NUMBERING



## 6.2 ACQUIRE Group/6.3 ASETup Group

### :ACQUIRE:RTOut:FILENAME

Function Sets the file name of auto naming of realtime recording or queries the current setting.

Syntax :ACQUIRE:RTOut:FILENAME <String>  
:ACQUIRE:RTOut:FILENAME?  
<String>=Up to 16 characters

Example :ACQUIRE:RTOut:FILENAME "ABC"  
:ACQUIRE:RTOut:FILENAME? ->  
:ACQUIRE:RTOut:FILENAME "ABC"

### :ACQUIRE:RTOut:COMMENT

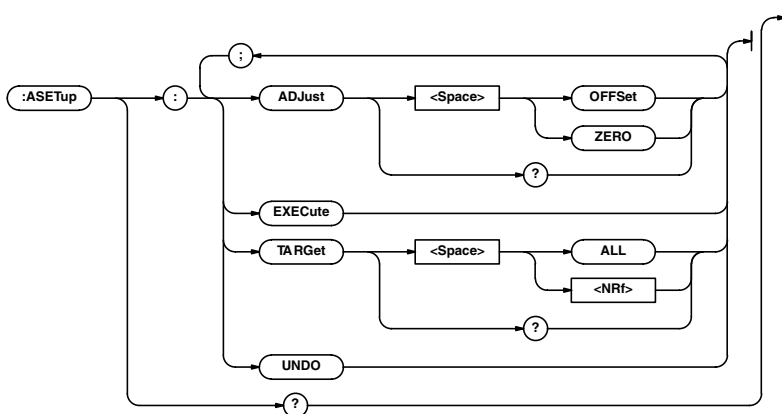
Function Sets the comment of realtime recording or queries the current setting.

Syntax :ACQUIRE:RTOut:COMMENT <String>  
:ACQUIRE:RTOut:COMMENT?  
<String>=Up to 160 characters

Example :ACQUIRE:RTOut:COMMENT "XYZ"  
:ACQUIRE:RTOut:COMMENT? ->  
:ACQUIRE:RTOut:COMMENT "XYZ"

## 6.3 ASETup Group

The commands in this group deal with the auto setup. You can make the same settings, execute the same operations, and make the same inquiries as when the SETUP key on the front panel is used.



### :ASETup?

Function Queries all settings related to auto setup.

Syntax :ASETup?

Example :ASETUP? -> :ASETUP:ADJUST ZERO;  
TARGET ALL

### :ASETup:ADJust

Function Sets the center position after auto setup or queries the current setting.

Syntax :ASETup:ADJust {OFFSet|ZERO}  
:ASETup:ADJust?

Example :ASETUP:ADJUST ZERO  
:ASETUP:ADJUST? -> :ASETUP:  
ADJUST ZERO

### :ASETup:EXECute

Function Executes auto setup.

Syntax :ASETup:EXECute

Example :ASETUP:EXECUTE

### :ASETup:TARGet

Function Sets the target channel for auto setup or queries the current setting.

Syntax :ASETup:TARGet {ALL|<NRf>}  
:ASETup:TARGet?  
<NRf>=1 to 16

Example :ASETUP:TARGET ALL  
:ASETUP:TARGET? -> :ASETUP:  
TARGET ALL

Description An error occurs if you specify a channel in which a module is not installed.

### :ASETup:UNDO

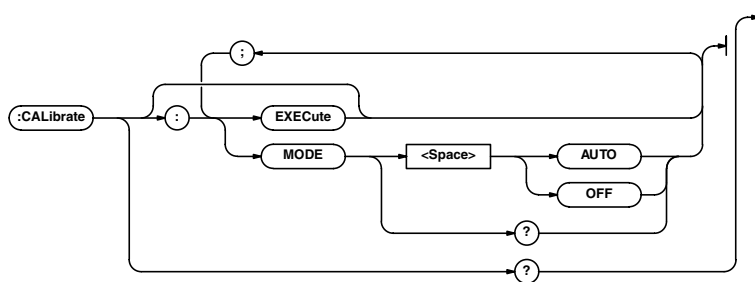
Function Cancels auto setup.

Syntax :ASETup:UNDO

Example :ASETUP:UNDO

## 6.4 CALibrate Group

The commands in this group deal with calibrations. You can make the same settings, execute the same operations, and make the same inquiries as when the SHIFT+SETUP key on the front panel is used.



### **:CALibrate?**

Function Queries all settings related to the calibration.  
 Syntax :CALibrate?  
 Example :CALIBRATE? -> :CALIBRATE:MODE AUTO

### **:CALibrate[:EXECute]**

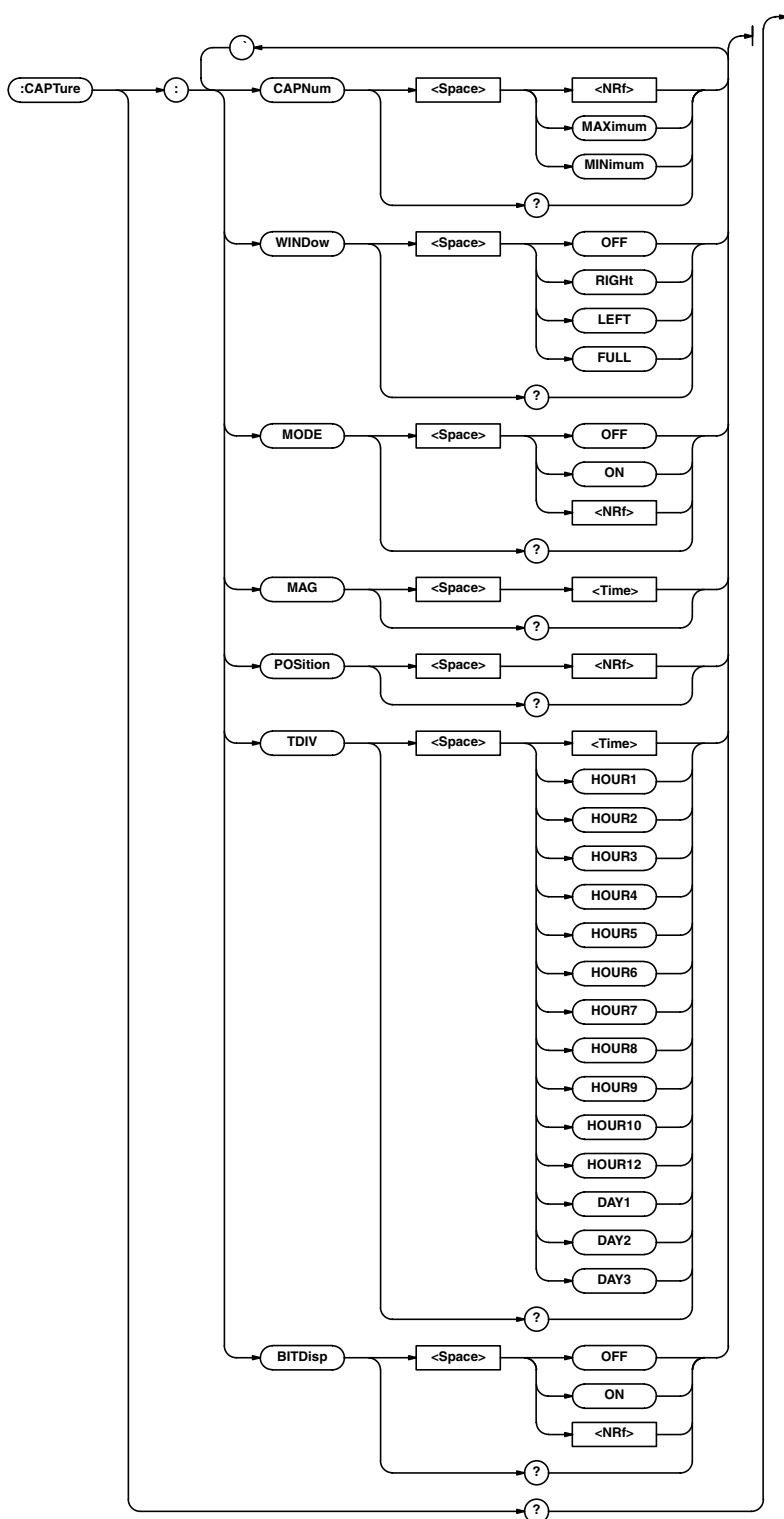
Function Executes the calibration.  
 Syntax :CALibrate[:EXECute]  
 Example :CALIBRATE:EXECUTE

### **:CALibrate:MODE**

Function Turns ON/OFF the auto calibration or queries the current setting.  
 Syntax :CALibrate:MODE {AUTO|OFF}  
 :CALibrate?  
 Example :CALIBRATE:MODE AUTO  
 :CALIBRATE? -> :CALIBRATE:MODE AUTO

## 6.5 CAPTure Group

The commands in this group deal with the dual capture function. You can make the same settings and inquiries as when the DUAL CAPTURE key on the front panel is used.



**:CAPTURE?**

Function Queries all settings related to dual capture.  
 Syntax :CAPTURE?  
 Example :CAPTURE? -> :CAPTURE:MODE 0;  
 TDIV 100.0E-06;WINDOW OFF;  
 MAG 100.0E-06;POSITION 0.0000000;  
 CAPNUM 0;BITDISP 1

**:CAPTURE:BITDISP**

Function Sets whether to display the dual capture event bit or queries the current setting.  
 Syntax :CAPTURE:BITDISP {<Boolean>}  
 :CAPTURE:BITDISP?  
 Example :CAPTURE:BITDISP 1  
 :CAPTURE:BITDISP? -> :CAPTURE:BITDISP 1

**:CAPTURE:MODE**

Function Sets whether or not to perform dual capture or queries the current setting.  
 Syntax :CAPTURE:MODE {<Boolean>}  
 :CAPTURE:MODE?  
 Example :CAPTURE:MODE 1  
 :CAPTURE:MODE? -> :CAPTURE:MODE 1

**:CAPTURE:TDIV**

Function Sets the T/div value of the capture or queries the current setting.  
 Syntax :CAPTURE:TDIV {<Time>}  
 :CAPTURE:TDIV?  
 <Time>=500 ns to 60 s  
 Example :CAPTURE:TDIV 1US  
 :CAPTURE:TDIV? -> :CAPTURE:TDIV 1.000E-06

**:CAPTURE:WINDOW**

Function Sets the size of the sub waveform window or queries the current setting.  
 Syntax :CAPTURE:WINDOW  
 {OFF|RIGHT|LEFT|FULL}  
 Example :CAPTURE:WINDOW RIGHT  
 :CAPTURE:WINDOW? -> :CAPTURE:WINDOW RIGHT

**:CAPTURE:POSITION**

Function Sets the horizontal position of the sub waveform window or queries the current setting.  
 Syntax :CAPTURE:POSITION {<NRf>}  
 :CAPTURE:POSITION?  
 <NRf>=-5 to 5 divisions (10 divisions/displayed record length steps)  
 Example :CAPTURE:POSITION 3  
 :CAPTURE:POSITION? -> :CAPTURE:POSITION 3.0000000

**:CAPTURE:MAG**

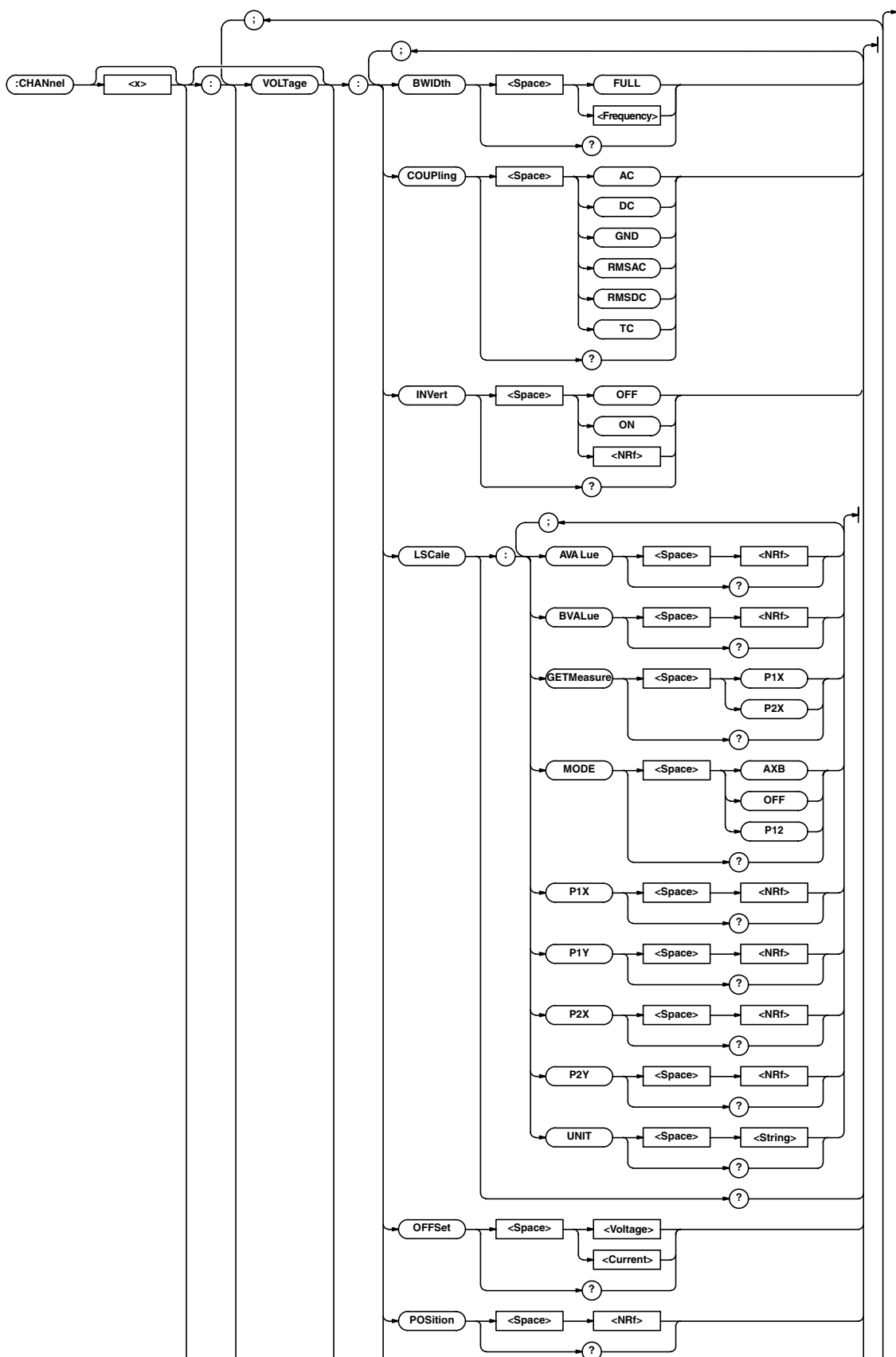
Function Sets the zoom rate of the sub waveform window or queries the current setting.  
 Syntax :CAPTURE:MAG {<Time>}  
 :CAPTURE:MAG?  
 <Time>=500 ns to 60 s  
 Example :CAPTURE:MAG 1MS  
 :CAPTURE:MAG 1MS  
 :CAPTURE:MAG? -> :CAPTURE:MAG 100.0E-06

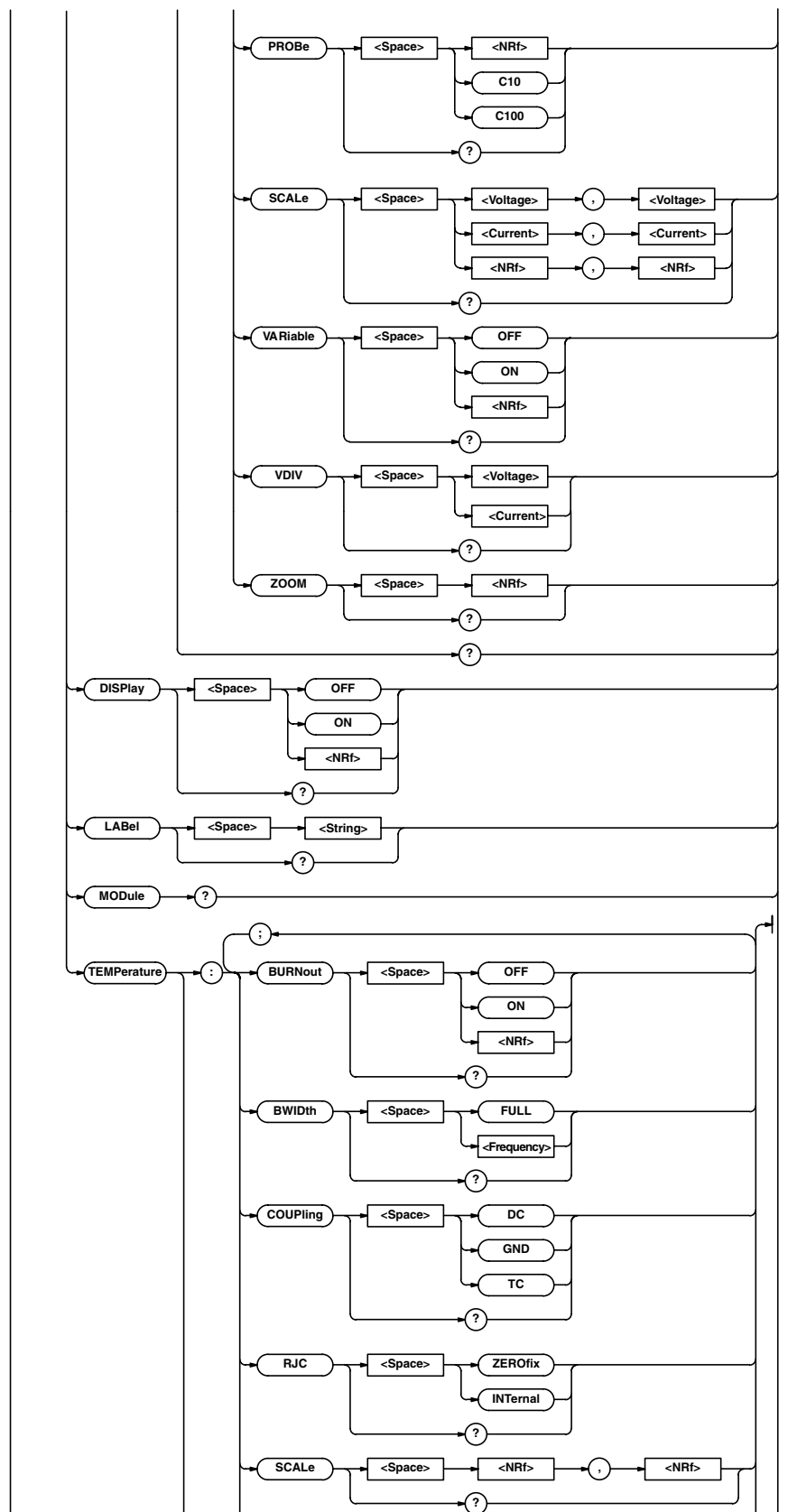
**:CAPTURE:CAPNUM**

Function Sets the dual capture number or queries the current setting.  
 Syntax :CAPTURE:CAPNUM  
 {<NRf>|MAXimum|MINimum}  
 :CAPTURE:CAPNUM?  
 Example :CAPTURE:CAPNUM -10  
 :CAPTURE:CAPNUM? -> :CAPTURE:CAPNUM -10

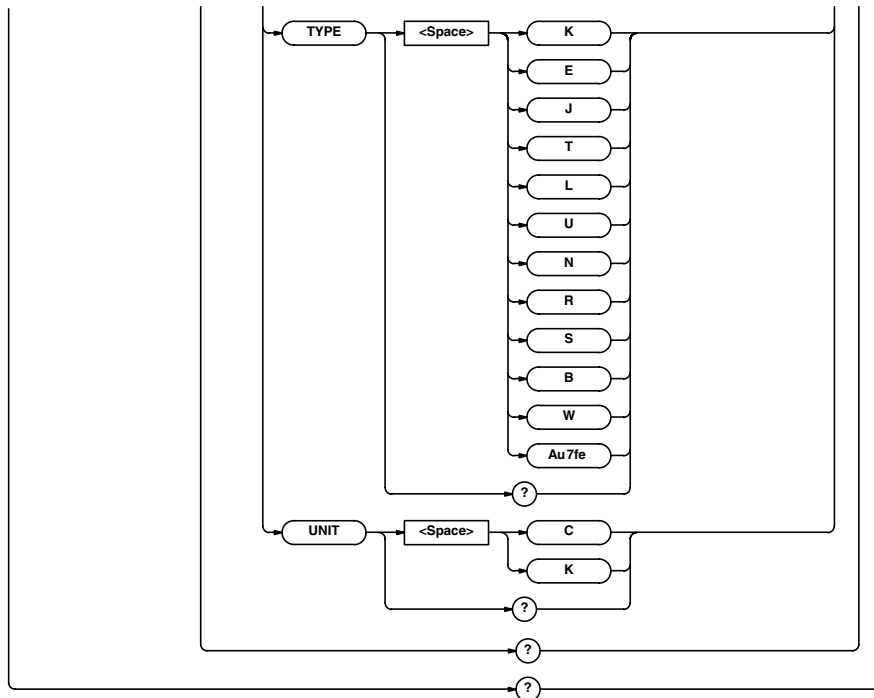
### 6.6 CHANnel Group

The commands in this group deal with the vertical axis of each channel. You can make the same settings and inquiries as when the VERTICAL group (CH key and V/DIV key) on the front panel is used.





## 6.6 CHANnel Group



### :CHANnel<x>?

**Function** Queries all settings related to the vertical axis of the channel.

**Syntax** :CHANnel<x>?  
<x>=1 to 16

**Example** (The following is an example when a voltage module is installed in channel 1.)  

```
:CHANNEL1? -> :CHANNEL1:DISPLAY 1;
VOLTAGE:VDIV 50.000E+00;VARIABLE 0;
COUPLING DC;PROBE 10;BWIDTH FULL;
SCALE 250.000E+00,-250.000E+00;
POSITION 0.00;LSCALE:MODE OFF;
AVALUE 1.00000E+00;
BVALUE 0.00000E+00;
P1X 0.00000E+00;P1Y 0.00000E+00;
P2X 1.00000E+00;P2Y 1.00000E+00;
UNIT "";:CHANNEL1:VOLTAGE:
ZOOM 1.000;INVERT 0;
OFFSET 0.0E+00;:CHANNEL1:
TEMPERATURE:COUPLING DC;:CHANNEL1:
LABEL "CH1"
```

**Description** An error occurs, if a module is not installed in the channel (slot).

### :CHANnel<x>:DISPlay

**Function** Turns ON/OFF the waveform display of the channel or queries the current setting.

**Syntax** :CHANnel<x>:DISPlay {<Boolean>}  
:CHANnel<x>:DISPlay?  
<x> = 1 to 16

**Example** :CHANNEL1:DISPLAY ON  
:CHANNEL1:DISPLAY? -> :CHANNEL1:DISPLAY 1

**Description** An error occurs, if a module is not installed in the channel (slot).

### :CHANnel<x>:LABel

**Function** Sets the waveform label of channel or queries the current setting.

**Syntax** :CHANnel<x>:LABel {<String>}  
:CHANnel<x>:LABel?  
<x>=1 to 16  
<String>=Up to 8 characters

**Example** :CHANNEL1:LABEL "TRACE1"  
:CHANNEL1:LABEL? -> :CHANNEL1:LABEL "TRACE1"

**Description**

- Only the characters and symbols displayed on the keyboard on the screen can be used.
- In ASCII code Ω and !sm are expressed as 1EH and 1FH, respectively.
- An error occurs, if a module is not installed in the channel (slot).

### :CHANnel<x>:MODUle?

**Function** Queries the module installed in the channel (slot).

**Syntax** :CHANnel<x>:MODUle?  
<x>=1 to 16

**Example** :CHANNEL1:MODULE? -> :CHANNEL1:MODULE M701250

**Description** The following values are returned from the module.

NOMODULE	No module
M701250	701250 (HS10M12)
M701251	701251 (HS1M16)
M701265	701265 (TEMP/HPV)

**:CHANnel<x>:TEMPerature?**

**Function** Queries all settings when the Temperature, High Precision Voltage Isolation Module is installed in the channel (slot).

**Syntax** :CHANnel<x>:TEMPerature?  
<x>=1 to 16

**Example** :CHANNEL1:TEMPERATURE? ->  
:CHANNEL1:TEMPERATURE:COUPLING TC;  
TYPE K;BWIDTH FULL;  
SCALE 1300.0,-200.0;UNIT C;RJC 1;  
BURNOUT 0

**Description** An error occurs if the Temperature, High Precision Voltage Isolation Module is not installed.

**:CHANnel<x>:TEMPerature:BURNout**

**Function** Sets whether to detect burnouts when the Temperature, High Precision Voltage Isolation Module is installed in the channel (slot) or queries the current setting.

**Syntax** :CHANnel<x>:TEMPerature:BURNout  
{<Boolean>}

**Example** :CHANNEL1:TEMPERATURE:BURNOUT ON  
:CHANNEL1:TEMPERATURE:BURNOUT? ->  
:CHANNEL1:TEMPERATURE:BURNOUT 1

**Description** An error occurs if the Temperature, High Precision Voltage Isolation Module is not installed.

**:CHANnel<x>:TEMPerature:BWIDth**

**Function** Sets the filter when the Temperature, High Precision Voltage Isolation Module is installed in the channel (slot) or queries the current setting.

**Syntax** :CHANnel<x>:TEMPerature:BWIDth  
{FULL|<Frequency>}  
:CHANnel<x>:TEMPerature:BWIDth?  
<x>=1 to 16  
<Frequency>=2, 8, 30 (Hz)

**Example** :CHANNEL1:TEMPERATURE:BWIDTH 2.0HZ  
:CHANNEL1:TEMPERATURE:BWIDTH? ->  
:CHANNEL1:TEMPERATURE:  
BWIDTH 2.0E+00

**Description** An error occurs if the Temperature, High Precision Voltage Isolation Module is not installed.

**:CHANnel<x>:TEMPerature:COUPLing**

**Function** Sets the input coupling when the Temperature, High Precision Voltage Isolation Module is installed in the channel (slot) or queries the current setting.

**Syntax** :CHANnel<x>:TEMPerature:COUPLing  
{TC|DC|GND}  
:CHANnel<x>:TEMPerature:COUPLing?  
<x>=1 to 16

**Example** :CHANNEL1:TEMPerature:COUPLING DC  
:CHANNEL1:TEMPerature:COUPLING? ->  
:CHANNEL1:TEMPerature:COUPLING DC

**Description** An error occurs if the Temperature, High Precision Voltage Isolation Module is not installed.

**:CHANnel<x>:TEMPerature:RJC**

**Function** Sets the RJC when the Temperature, High Precision Voltage Isolation Module is installed in the channel (slot) or queries the current setting.

**Syntax** :CHANnel<x>:TEMPerature:RJC  
{<Boolean>}  
:CHANnel<x>:TEMPerature:RJC?  
<x>=1 to 16

**Example** :CHANNEL1:TEMPERATURE:RJC ON  
:CHANNEL1:TEMPERATURE:RJC? ->  
:CHANNEL1:TEMPERATURE:RJC 1

**Description** An error occurs if the Temperature, High Precision Voltage Isolation Module is not installed.

**:CHANnel<x>:TEMPerature:SCALE**

**Function** Sets the upper and lower limits on the screen when the Temperature, High Precision Voltage Isolation Module is installed in the channel (slot) or queries the current setting.

**Syntax** :CHANnel<x>:TEMPerature:SCALE  
{<Nrf>,<Nrf>}  
:CHANnel<x>:TEMPerature:SCALE?  
<x>=1 to 16  
<Nrf>=-5432 to 5432 (0.1°C steps)

**Example** :CHANNEL1:TEMPERATURE:  
SCALE 1200.0,-200.0  
:CHANNEL1:TEMPERATURE:SCALE? ->  
:CHANNEL1:TEMPERATURE:  
SCALE 1200.0,-200.0

**Description**

- The unit of the value is the unit specified by :CHANnel<x>:TEMPerature:UNIT.
- An error occurs if the Temperature, High Precision Voltage Isolation Module is not installed.

**:CHANnel<x>:TEMPerature:TYPE**

**Function** Sets the thermocouple type to be used when the Temperature, High Precision Voltage Isolation Module is installed in the channel (slot) or queries the current setting.

**Syntax** :CHANnel<x>:TEMPerature:TYPE  
{K|E|J|T|L|U|N|R|S|B|W|Au7fe}  
:CHANnel<x>:TEMPerature:TYPE?  
<x>=1 to 16

**Example** :CHANNEL1:TEMPERATURE:TYPE K  
:CHANNEL1:TEMPERATURE:TYPE? ->  
:CHANNEL1:TEMPERATURE:TYPE K

**Description** An error occurs if the Temperature, High Precision Voltage Isolation Module is not installed.



## 6.6 CHANnel Group

### :CHANnel<x>:TEMPerature:UNIT

**Function** Sets the unit of the upper and lower limits when the Temperature, High Precision Voltage Isolation Module is installed in the channel (slot) or queries the current setting.

**Syntax** :CHANnel<x>:TEMPerature:UNIT  
{C|F|K}  
:CHANnel<x>:TEMPerature:UNIT?  
<x>=1 to 16

**Example** :CHANNEL1:TEMPERATURE:UNIT C  
:CHANNEL1:TEMPERATURE:UNIT? ->  
:CHANNEL1:TEMPERATURE:UNIT C

**Description** An error occurs if the Temperature, High Precision Voltage Isolation Module is not installed.

### :CHANnel<x>:VOLTage?

**Function** Queries all settings when the voltage module is installed in the channel (slot).

**Syntax** :CHANnel<x>:VOLTage?  
<x>=1 to 16

**Example** :CHANNEL1:VOLTAGE? -> :CHANNEL1:  
VOLTAGE:VDIV 50.0000E+00;VARIABLE 0;  
COUPLING DC;PROBE 10;BWIDTh FULL;  
SCALE 250.0000E+00,-250.0000E+00;  
POSITION 0.00;LSCALE:MODE OFF;  
AVALUE 1.000000E+00;  
BVALUE 0.000000E+00;  
P1X 0.000000E+00;P1Y 0.000000E+00;  
P2X 1.000000E+00;P2Y 1.000000E+00;  
UNIT "";:CHANNEL1:VOLTAGE:  
ZOOM 1.000;INVERT 0;OFFSET 0.00E+00

**Description** An error occurs if the voltage module is not installed.

### :CHANnel<x>[:VOLTage]:BWIDth

**Function** Sets the filter when the voltage module is installed in the channel (slot) or queries the current setting.

**Syntax** :CHANnel<x>[:VOLTage]:BWIDth  
{FULL|<Frequency>}  
:CHANnel<x>[:VOLTage]:BWIDth?  
<x>=1 to 16  
<Frequency>=500 Hz, 5 kHz, 50 kHz, or 500 kHz (for the M701250)  
400 Hz, 4 kHz, or 40 kHz (for the M701251)  
2 Hz, 8 Hz, or 30 Hz (for the M701265)

**Example** :CHANNEL1:VOLTAGE:BWIDTh FULL  
:CHANNEL1:VOLTAGE:BWIDTh? ->  
:CHANNEL1:VOLTAGE:BWIDTh FULL

**Description** An error occurs if the voltage module is not installed.

### :CHANnel<x>[:VOLTage]:COUPLing

**Function** Sets the input coupling when the voltage module is installed in the channel (slot) or queries the current setting.

**Syntax** :CHANnel<x>[:VOLTage]:COUPLing  
{AC|DC|GND|TC}  
:CHANnel<x>[:VOLTage]:COUPLing?  
<x>=1 to 16

**Example** :CHANNEL1:VOLTAGE:COUPLING DC  
:CHANNEL1:VOLTAGE:COUPLING? ->  
:CHANNEL1:VOLTAGE:COUPLING DC

**Description**

- An error occurs if the voltage module is not installed.
- The "TC" setting is not available on the M701250/M701251.
- The "AC" setting is not available on the M701265.

### :CHANnel<x>[:VOLTage]:INVert

**Function** Sets whether to invert the waveform when the voltage module is installed in the channel (slot) or queries the current setting.

**Syntax** :CHANnel<x>[:VOLTage]:INVert  
{<Boolean>}  
:CHANnel<x>[:VOLTage]:INVert?  
<x>=1 to 16

**Example** :CHANNEL1:VOLTAGE:INVERT ON  
:CHANNEL1:VOLTAGE:INVERT? ->  
:CHANNEL1:VOLTAGE:INVERT 1

**Description** An error occurs if the voltage module is not installed.

### :CHANnel<x>[:VOLTage]:LSCale?

**Function** Queries all settings related to linear scaling when the voltage module is installed in the channel (slot).

**Syntax** :CHANnel<x>[:VOLTage]:LSCale?  
<x>=1 to 16

**Example** :CHANNEL1:VOLTAGE:LSCALE? ->  
:CHANNEL1:VOLTAGE:LSCALE:MODE P12;  
AVALUE 1.000000E+00;  
BVALUE 0.000000E+00;  
P1X 0.000000E+00;P1Y 0.000000E+00;  
P2X 1.000000E+00;P2Y 1.000000E+00;  
UNIT "V"

**Description** An error occurs if the voltage module is not installed.

### :CHANnel<x>[:VOLTage]:LSCale:AVALue

**Function** Sets scaling coefficient A of linear scaling when the voltage module is installed in the channel (slot) or queries the current setting.

**Syntax** :CHANnel<x>[:VOLTage]:LSCale:AVALue  
{<NRf>}  
:CHANnel<x>[:VOLTage]:LSCale:  
AVALue?  
<x>=1 to 16  
<NRf>=-9.9999E+30 to 9.9999E+30

**Example** :CHANNEL1:VOLTAGE:LSCALE:AVALUE 10  
:CHANNEL1:VOLTAGE:LSCALE:AVALUE? ->  
:CHANNEL1:VOLTAGE:LSCALE:  
AVALUE 10.0000E+00

**Description** An error occurs if the voltage module is not installed.

**:CHANnel<x>[:VOLTage]:LSCale:BVALue**  
 Function Sets offset B of linear scaling when the voltage module is installed in the channel (slot) or queries the current setting.  
 Syntax :CHANnel<x>[:VOLTage]:LSCale:BVALue {<NRf>}  
 :CHANnel<x>[:VOLTage]:LSCale:BVALue?  
 <x>=1 to 16  
 <NRf>=-9.9999E+30 to 9.9999E+30  
 Example :CHANNEL1:VOLTAGE:LSCALE:BVALUE 10  
 :CHANNEL1:VOLTAGE:LSCALE:BVALUE? ->  
 :CHANNEL1:VOLTAGE:LSCALE:BVALUE 10.0000E+00  
 Description An error occurs if the voltage module is not installed.

**:CHANnel<x>[:VOLTage]:LSCale:GETMeasure**  
 Function Executes the measurement of the P1:X|P2:X values of linear scaling when the voltage module is installed.  
 Syntax :CHANnel<x>[:VOLTage]:LSCale:GETMeasure {P1X|P2X}  
 <x>=1 to 16  
 Example :CHANNEL1:VOLTAGE:GETMeasure P1X  
 Description An error occurs if the voltage module is not installed.

**:CHANnel<x>[:VOLTage]:LSCale:MODE**  
 Function Sets the linear scaling function when the voltage module is installed in the channel (slot) or queries the current setting.  
 Syntax :CHANnel<x>[:VOLTage]:LSCale:MODE {AXB|OFF|P12}  
 :CHANnel<x>[:VOLTage]:LSCale:MODE?  
 <x>=1 to 16  
 Example :CHANNEL1:VOLTAGE:LSCALE:MODE AXB  
 :CHANNEL1:VOLTAGE:LSCALE:MODE? ->  
 :CHANNEL1:VOLTAGE:LSCALE:MODE AXB  
 Description An error occurs if the voltage module is not installed.

**:CHANnel<x>[:VOLTage]:LSCale:{P1X|P1Y|P2X|P2Y}**  
 Function Sets the P1:X|P1:Y|P2:X|P2:Y values of linear scaling when the voltage module is installed in the channel (slot) or queries the current setting.  
 Syntax :CHANnel<x>[:VOLTage]:LSCale:{P1X|P1Y|P2X|P2Y} {<NRf>}  
 :CHANnel<x>[:VOLTage]:LSCale:{P1X|P1Y|P2X|P2Y}?  
 <x>=1 to 16  
 <NRf>=-9.9999E+30 to 9.9999E+30  
 Example :CHANNEL1:VOLTAGE:LSCALE:P1X 10  
 :CHANNEL1:VOLTAGE:LSCALE:P1X? ->  
 :CHANNEL1:VOLTAGE:LSCALE:P1X 10.0000E+00  
 Description An error occurs if the voltage module is not installed.

**:CHANnel<x>[:VOLTage]:LSCale:UNIT**  
 Function Sets the unit to be added to the result of linear scaling when the voltage module is installed in the channel (slot) or queries the current setting.  
 Syntax :CHANnel<x>[:VOLTage]:LSCale:UNIT {<String>}  
 :CHANnel<x>[:VOLTage]:LSCale:UNIT?  
 <x>=1 to 16  
 <String>=Up to 4 characters  
 Example :CHANNEL1:VOLTAGE:LSCALE:UNIT "RPM"  
 :CHANNEL1:VOLTAGE:LSCALE:UNIT? ->  
 :CHANNEL1:VOLTAGE:LSCALE:UNIT "RPM"  
 Description An error occurs if the voltage module is not installed.

**:CHANnel<x>[:VOLTage]:OFFSet**  
 Function Sets the offset voltage when the voltage module is installed in the channel (slot) or queries the current setting.  
 Syntax :CHANnel<x>[:VOLTage]:OFFSet {<Voltage>|<Current>}  
 :CHANnel<x>[:VOLTage]:OFFSet?  
 <x>=1 to 16  
 <Voltage>,<Current>=The selectable range varies depending on the range setting. For details, see the DL750 User's Manual.  
 Example :CHANNEL1:VOLTAGE:OFFSET 5V  
 :CHANNEL1:VOLTAGE:OFFSET? ->  
 :CHANNEL1:VOLTAGE:OFFSET 5.0000E+00  
 Description An error occurs if the voltage module is not installed.

**:CHANnel<x>[:VOLTage]:POSition**  
 Function Sets the vertical position when the voltage module is installed in the channel (slot) or queries the current setting.  
 Syntax :CHANnel<x>[:VOLTage]:POSition {<NRf>}  
 :CHANnel<x>[:VOLTage]:POSition?  
 <x>=1 to 16  
 <NRf>=-5.00 to 5.00 (div, 0.01 div steps)  
 Example :CHANNEL1:VOLTAGE:POSITION 1.00  
 :CHANNEL1:VOLTAGE:POSITION? ->  
 :CHANNEL1:VOLTAGE:POSITION 1.00  
 Description An error occurs if the voltage module is not installed.

**:CHANnel<x>[:VOLTage]:PROBe**  
 Function Sets the probe type when the voltage module is installed in the channel (slot) or queries the current setting.  
 Syntax :CHANnel<x>[:VOLTage]:PROBe {<NRf>|C10|C100}  
 :CHANnel<x>[:VOLTage]:PROBe?  
 <x>=1 to 16  
 <NRf>=1, 10, 100, or 1000  
 Example :CHANNEL1:VOLTAGE:PROBE 10  
 :CHANNEL1:VOLTAGE:PROBE? ->  
 :CHANNEL1:VOLTAGE:PROBE 10  
 Description
 

- An error occurs if the voltage module is not installed.
- This command is not available on the M701265.

## 6.6 CHANnel Group/6.7 CLear Group

### :CHANnel<x>[:VOLTage]:SCALE

Function	Sets the upper and lower limits on the screen when the voltage module is installed in the channel (slot) or queries the current setting.
Syntax	:CHANnel<x>[:VOLTage]:SCALE {<Voltage>,<Current> <Current>,<Current> <NRf>,<NRf>} :CHANnel<x>[:VOLTage]:SCALE? <x>=1 to 16 <Voltage>,<Current>,<NRf>=The selectable range varies depending on the range setting. For details, see the <i>DL750 User's Manual</i> .
Example	:CHANNEL1:VOLTAGE:SCALE 20V,-20V :CHANNEL1:VOLTAGE:SCALE? -> :CHANNEL1:VOLTAGE: SCALE 20.000E+00,-20.000E+00
Description	An error occurs if the voltage module is not installed.

### :CHANnel<x>[:VOLTage]:VARIABLE

Function	Sets whether to enable Variable mode when the voltage module is installed in the channel (slot) or queries the current setting.
Syntax	:CHANnel<x>[:VOLTage]:VARIABLE {<Boolean>} :CHANnel<x>[:VOLTage]:VARIABLE? <x>=1 to 16
Example	:CHANNEL1:VOLTAGE:VARIABLE ON :CHANNEL1:VOLTAGE:VARIABLE? -> :CHANNEL1:VOLTAGE:VARIABLE 1

### :CHANnel<x>[:VOLTage]:VDIV

Function	Sets the V/div value when the voltage module is installed in the channel (slot) or queries the current setting.
Syntax	:CHANnel<x>[:VOLTage]:VDIV {<Voltage> <Current>} :CHANnel<x>[:VOLTage]:VDIV? <x>=1 to 16 <Voltage>=5 mV to 20 V (M701250 when the probe attenuation is 1:1) 1 mV to 20 V (M701251 when the probe attenuation is 1:1) 0.1 mV to 10 V (M701265)
Example	:CHANNEL1:VOLTAGE:VDIV 5V :CHANNEL1:VOLTAGE:VDIV? -> :CHANNEL1:VOLTAGE:VDIV 5.000E+00
Description	An error occurs if the voltage module is not installed.

### :CHANnel<x>[:VOLTage]:ZOOM

Function	Sets the vertical zoom rate when the voltage module is installed in the channel (slot) or queries the current setting.
Syntax	:CHANnel<x>[:VOLTage]:ZOOM {<NRf>} :CHANnel<x>[:VOLTage]:ZOOM? <x>=1 to 16 <NRf>=0.1, 0.111, 0.125, 0.143, 0.167, 0.2, 0.25, 0.33, 0.4, 0.5, 0.556, 0.625, 0.667, 0.714, 0.8, 0.833, 1, 1.11, 1.25, 1.33, 1.43, 1.67, 2, 2.22, 2.5, 3.33, 4, 5, 6.67, 8, 10, 12.5, 16.7, 20, 25, 40, 50, or 100
Example	:CHANNEL1:VOLTAGE:ZOOM 10 :CHANNEL1:VOLTAGE:ZOOM? -> :CHANNEL1:VOLTAGE:ZOOM 10.000
Description	An error occurs if the voltage module is not installed.

## 6.7 CLear Group

The commands in this group executes clear trace. You can execute the same operation as the CLEAR key on the front panel.

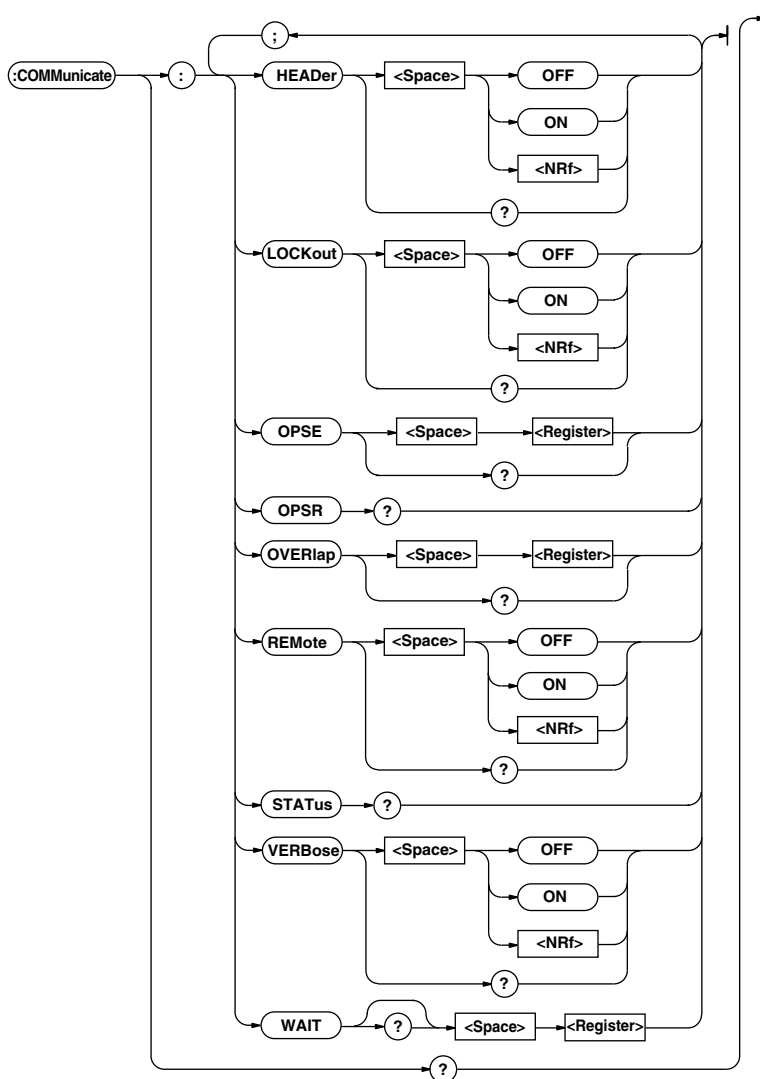


### :CLear

Function	Executes clear trace.
Syntax	:CLear
Example	:CLEAR
Description	Snapshot is carried out using "SNAP."

## 6.8 COMMunicate Group

The commands in this group deal with communications. There are no front panel keys that correspond to the commands in this group.



### :COMMunicate?

Function Queries all settings related to communications.

Syntax :COMMunicate?

Example :COMMUNICATE? ->:COMMUNICATE:  
 HEADER 1;OPSE 352;OVERLAP 352;  
 VERBOSE 1

### :COMMunicate:HEADer

Function Sets whether to add a header to the response to a query (example CHANNEL1:VOLTAGE:PROBE 10) or not add the header (example 10).

Syntax :COMMunicate:HEADer {<Boolean>}  
 :COMMunicate:HEADer?

Example :COMMUNICATE:HEADER ON  
 :COMMUNICATE:HEADER? ->  
 :COMMUNICATE:HEADER 1

### :COMMunicate:LOCKout

Function Sets or clears local lockout.

Syntax :COMMunicate:LOCKout {<Boolean>}  
 :COMMunicate:LOCKout?

Example :COMMUNICATE:LOCKOUT ON  
 :COMMUNICATE:LOCKOUT? ->  
 :COMMUNICATE:LOCKOUT 1

Description This command is for the serial (RS-232), USB, or Ethernet interface. For GP-IB, this function is provided as an interface message.

## 6.8 COMMunicate Group

### :COMMunicate:OPSE (Operation Pending Status Enable register)

**Function** Sets the overlap command that is to be used by the \*OPC, \*OPC?, and \*WAI commands or queries the current setting.

**Syntax** :COMMunicate:OPSE <Register>  
:COMMunicate:OPSE?  
<Register>=0 to 65535, :COMMunicate:WAIT?  
See the command diagram.

**Example** :COMMUNICATE:OPSE 65535  
:COMMUNICATE:OPSE? -> :COMMUNICATE:OPSE 352

**Description** In the above example, all bits are set to 1 to make all overlap commands applicable. However, bits fixed to 0 are not set to 1. Thus, the response to the query indicates 1 for bits 5, 6, and 8 only.

### :COMMunicate:OPSR? (Operation Pending Status Register)

**Function** Queries the value of the operation pending status register.

**Syntax** :COMMunicate:OPSR?

**Example** :COMMUNICATE:OPSR? -> 0

**Description** For details on the operation pending status register, see the figure for the :COMMunicate:WAIT? command.

### :COMMunicate:OVERlap

**Function** Sets the commands to operate as overlap commands or queries the current setting.

**Syntax** :COMMunicate:OVERlap <Register>  
:COMMunicate:OVERlap?  
<Register>=0 to 65535,  
:COMMunicate:WAIT?  
See the command diagram.

**Example** :COMMUNICATE:OVERLAP 65535  
:COMMUNICATE:OVERLAP? ->  
:COMMUNICATE:OVERLAP 352

**Description**

- In the above example, all bits are set to 1 to make all overlap commands applicable. However, bits fixed to 0 are not set to 1. Thus, the response to the query indicates 1 for bits 5, 6, and 8 only.
- For the description of how to synchronize the program using COMMunicate:OVERlap, see page 5-8.
- In the above example, bits 5, 6, and 8 are set to 1 to make all overlap commands applicable (see the figure for the :COMMunicate:WAIT? command).

### :COMMunicate:REMOte

**Function** Sets remote or local. Remote mode is enabled when ON.

**Syntax** :COMMunicate:REMOte {<Boolean>}  
:COMMunicate:REMOte?

**Example** :COMMUNICATE:REMOTE ON  
:COMMUNICATE:REMOTE? ->  
:COMMUNICATE:REMOTE 1

**Description** This command is for the serial (RS-232), USB, or Ethernet interface. For GP-IB, this function is provided as an interface message.

### :COMMunicate:STATus?

**Function** Queries line-specific status.

**Syntax** :COMMunicate:STATus?

**Example** :COMMUNICATE:STATUS? ->  
:COMMUNICATE:STATUS 0

**Description** The meaning of each status bit is as follows:

Bit	GP-IB	RS-232
0	Unrecoverable transmission error	Parity error
1	Always 0	Framing error
2	Always 0	Break character detected
3 or greater	Always 0	Always 0

For USB and Ethernet communications, 0 is always returned. The status bit is set when the corresponding cause occurs and cleared when it is read.

### :COMMunicate:VERBose

**Function** Sets whether to return the response to a query using full spelling (example CHANNEL1:VOLTAGE:PROBE 10) or using abbreviation (example CHAN:PROB 10).

**Syntax** :COMMunicate:VERBose {<Boolean>}  
:COMMunicate:VERBose?

**Example** :COMMUNICATE:VERBOSE ON  
:COMMUNICATE:VERBOSE? ->  
:COMMUNICATE:VERBOSE 1

### :COMMunicate:WAIT

**Function** Waits for one of the specified extended events to occur.

**Syntax** :COMMunicate:WAIT <Register>  
<Register>= 0 to 65535 (extended event register, see page 7-4.)

**Example** :COMMUNICATE:WAIT 65535

**Description** For the description of how to synchronize the program using :COMMunicate:WAIT, see page 5-9.

### :COMMunicate:WAIT?

**Function** Creates the response that is returned when the specified event occurs.

**Syntax** :COMMunicate:WAIT? <Register>  
<Register>= 0 to 65535 (extended event register, see page 7-4.)

**Example** :COMMUNICATE:WAIT? 65535 -> 1

Operation pending status register/overlap enable register

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

When bit 5 (PRN)=1:

Built-in printer operation not complete

When bit 6 (ACS)=1:

Access to the medium not complete.

When bit 8 (HST)=1:

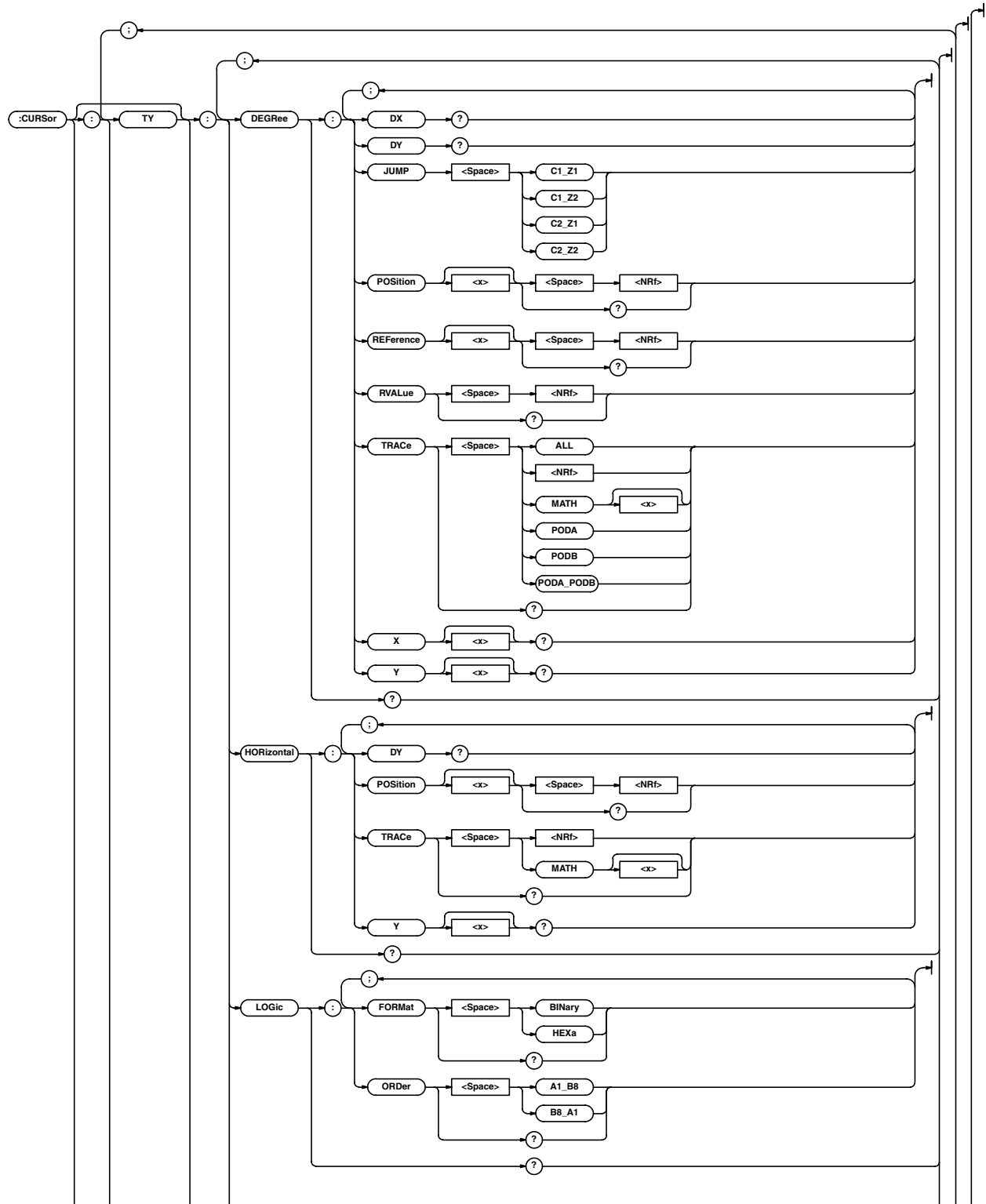
History search execution not complete

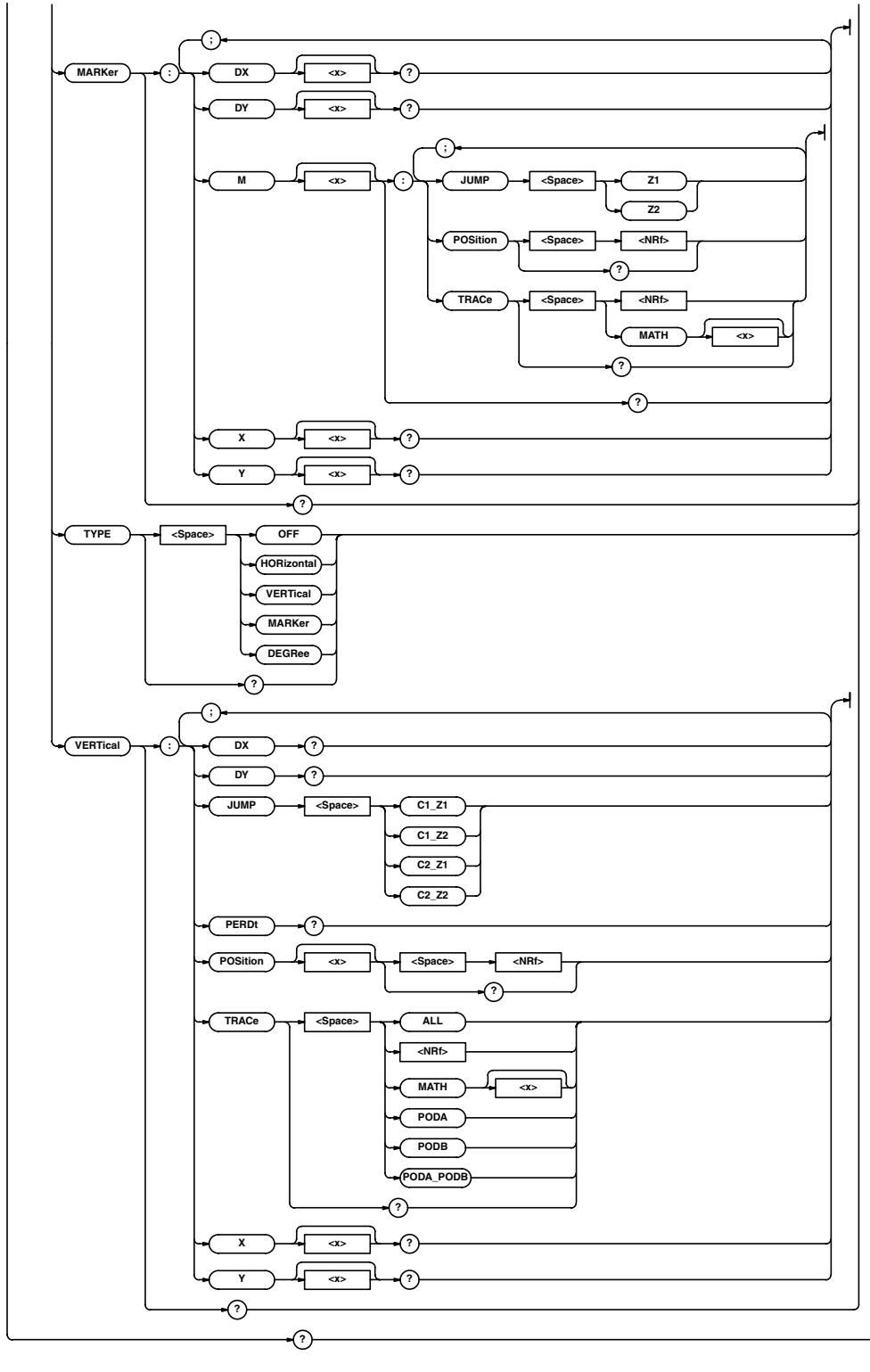
## 6.9 CURSOR Group

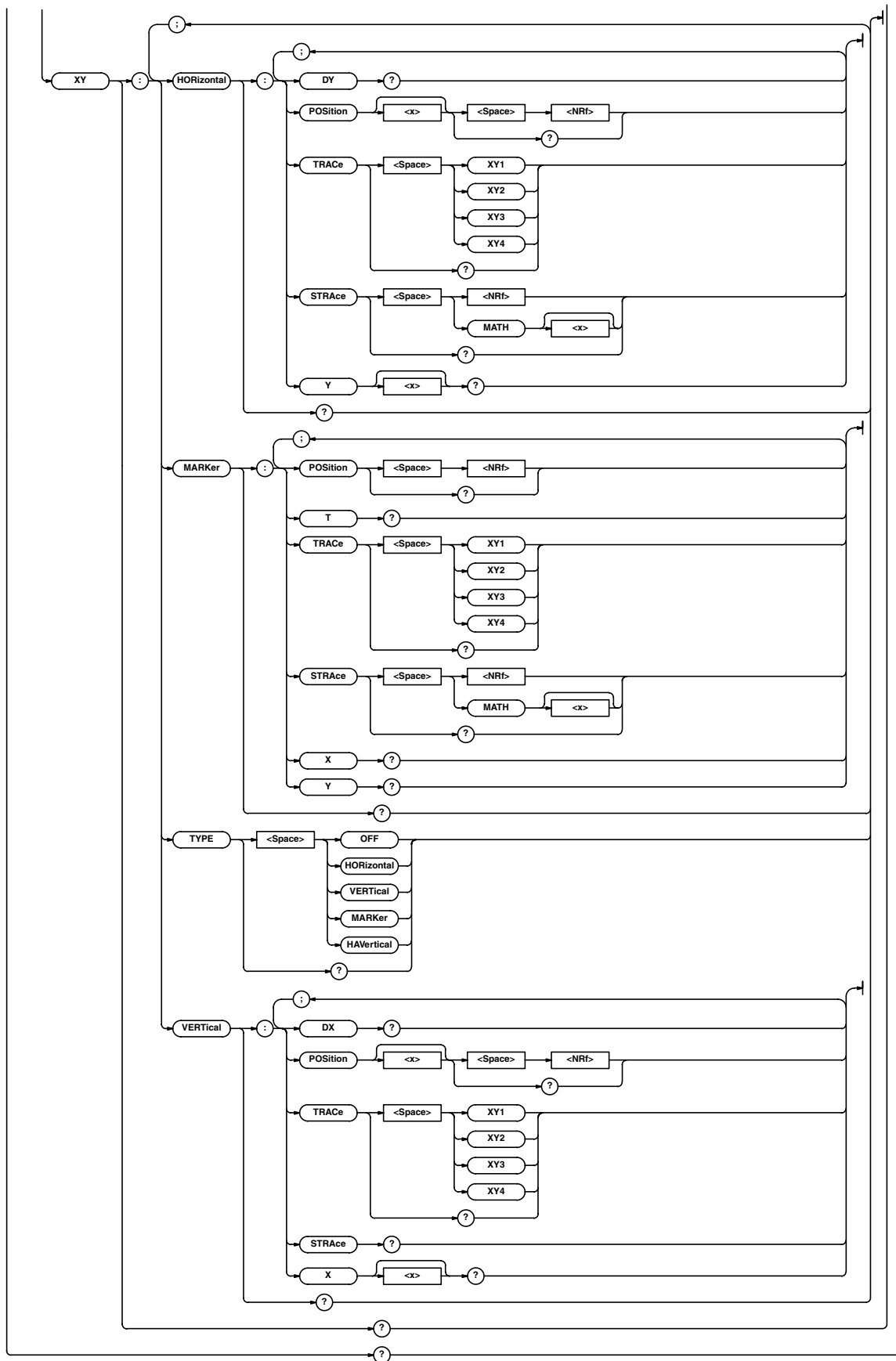
The commands in this group deal with cursor measurements. You can make the same settings and inquiries of settings and measured values as when CURSOR on the front panel is used.

When the time axis range setting is as follows, the selectable range varies depending on the record length setting. For details, see "Selectable Range of Cursor Position" in section 8.1 of the DL750 User's Manual.

<NRf>=-5 to 5 div









## 6.9 CURSor Group

### :CURSor?

**Function** Queries all settings related to cursor measurements.

**Syntax** :CURSOR?

**Example** :CURSOR? -> :CURSOR:TY:  
 TYPE HORIZONTAL;HORIZONTAL:TRACE 1;  
 POSITION1 3.00;POSITION2 -3.00;:  
 CURSOR:TY:VERTICAL:TRACE 1;  
 POSITION1 -4.0000000;  
 POSITION24.0000000;:CURSOR:TY:MARKER:  
 M1:TRACE 1;POSITION -3.0000000;:  
 CURSOR:TY:MARKER:M2:TRACE 2;  
 POSITION -1.0000000;:CURSOR:TY:  
 MARKER:M3:TRACE 3;  
 POSITION 1.0000000;:CURSOR:TY:  
 MARKER:M4:TRACE 4;  
 POSITION3.0000000;:CURSOR:TY:DEGREE:  
 TRACE 1;POSITION1 -4.0000000;  
 POSITION2 4.0000000;  
 REFERENCE1 -2.0000000;  
 REFERENCE2 2.0000000;RVALUE 360;:  
 CURSOR:TY:LOGIC:FORMAT BINARY;  
 ORDER A1\_B8;:CURSOR:XY:TYPE OFF;  
 HORIZONTAL:TRACE XY1;  
 POSITION1 3.00;POSITION2 -3.00;  
 STRACE 1;:CURSOR:XY:VERTICAL:  
 TRACE XY1;POSITION1 -3.00;  
 POSITION2 3.00;STRACE 1;:CURSOR:XY:  
 MARKER:TRACE XY1;  
 POSITION -3.0000000;STRACE 1

### :CURSor[:TY]?

**Function** Queries all settings related to the cursor of the T-Y display.

**Syntax** :CURSOR:TY?

**Example** :CURSOR:TY? -> :CURSOR:TY:  
 TYPE HORIZONTAL;HORIZONTAL:TRACE 1;  
 POSITION1 3.00;POSITION2 -3.00;:  
 CURSOR:TY:VERTICAL:TRACE 1;  
 POSITION1 -4.0000000;  
 POSITION24.0000000;:CURSOR:TY:  
 MARKER:M1:TRACE 1;  
 POSITION -3.0000000;:CURSOR:TY:  
 MARKER:M2:TRACE 2;  
 POSITION -1.0000000;:CURSOR:TY:  
 MARKER:M3:TRACE 3;  
 POSITION 1.0000000;:CURSOR:  
 TY:MARKER:M4:TRACE 4;  
 POSITION 3.0000000;:CURSOR:TY:  
 DEGREE:TRACE 1;  
 POSITION1 -4.0000000;  
 POSITION2 4.0000000;  
 REFERENCE1 -2.0000000;  
 REFERENCE2 2.0000000;RVALUE 360;:  
 CURSOR:TY:LOGIC:FORMAT BINARY;  
 ORDER A1\_B8

### :CURSor[:TY]:DEGREE?

**Function** Queries all settings related to the angle cursor of the T-Y display.

**Syntax** :CURSOR:TY:DEGREE?

**Example** :CURSOR:TY:DEGREE? -> :CURSOR:TY:  
 DEGREE:TRACE 1;  
 POSITION1 -4.0000000;  
 POSITION2 4.0000000;  
 REFERENCE1 -2.0000000;  
 REFERENCE2 2.0000000;RVALUE 360

### :CURSor[:TY]:DEGREE:DX?

**Function** Queries the angle difference !s!DD between angle cursors of the T-Y display.

**Syntax** :CURSOR:TY:DEGREE:DX?

**Example** :CURSOR:TY:DEGREE:DX? -> :CURSOR:  
 TY:DEGREE:DX 180.000000E+00

### :CURSor[:TY]:DEGREE:DY?

**Function** Queries the !s!DV value between angle cursors of the T-Y display.

**Syntax** :CURSOR:TY:DEGREE:DY?

**Example** :CURSOR:TY:DEGREE:DY? -> :CURSOR:  
 TY:DEGREE:DY 6.2500000E-03

### :CURSor[:TY]:DEGREE:JUMP

**Function** Jumps the angle cursor on the T-Y display onto the specified zoom waveform.

**Syntax** :CURSOR:TY:DEGREE:JUMP  
 {C1\_Z1|C1\_Z2|C2\_Z1|C2\_Z2}

**Example** :CURSOR:TY:DEGREE:JUMP C1\_Z1

**Description** Jumps to the center position of the zoom waveform.  
 C1 and C2 indicate Cursor1 and Cursor2, respectively.

### :CURSor[:TY]:DEGREE:POSITION<x>

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

**Syntax** :CURSOR:TY:DEGREE:POSITION<x>  
 {<Nrf>}  
 :CURSOR:TY:DEGREE:POSITION<x>?  
 <x>=1 or 2  
 <Nrf>=-5 to 5 divisions (10 divisions/displayed record length steps)

**Example** :CURSOR:TY:DEGREE:POSITION1 2  
 :CURSOR:TY:DEGREE:POSITION1? ->  
 :CURSOR:TY:DEGREE:  
 POSITION1 2.0000000

### :CURSor[:TY]:DEGREE:REFERENCE<x>

**Function** Sets the position of the start point (Reference 1) or the end point (Reference 2) of the angle reference of the T-Y display or queries the current setting.

**Syntax** :CURSOR:TY:DEGREE:REFERENCE<x>  
 {<Nrf>}  
 :CURSOR:TY:DEGREE:REFERENCE<x>?  
 <x>=1 or 2  
 <Nrf>=-5 to 5 divisions (10 divisions/displayed record length steps)

**Example** :CURSOR:TY:DEGREE:REFERENCE1 -1  
 :CURSOR:TY:DEGREE:REFERENCE1? ->  
 :CURSOR:TY:DEGREE:  
 REFERENCE1 -1.0000000

### :CURSor[:TY]:DEGREE:RVALUE

**Function** Sets the angle reference width or queries the current setting.

**Syntax** :CURSOR:TY:DEGREE:RVALUE {<Nrf>}  
 :CURSOR:TY:DEGREE:RVALUE?  
 <Nrf>=1 to 720

**Example** :CURSOR:TY:DEGREE:RVALUE 180  
 :CURSOR:TY:DEGREE:RVALUE? ->  
 :CURSOR:TY:DEGREE:RVALUE 180

**:CURSOR[:TY]:DEGREE:TRACE**

**Function** Sets the target waveform of the angle cursor of the T-Y display or queries the current setting.

**Syntax** :CURSOR:TY:DEGREE:TRACE  
{<NRf>|MATH<x>|ALL|PODA|PODB|PODA\_PODB}  
:CURSOR:TY:DEGREE:TRACE?  
<NRf>=1 to 16  
<x>=1 to 8

**Example** :CURSOR:TY:DEGREE:TRACE 1  
:CURSOR:TY:DEGREE:TRACE? ->  
:CURSOR:TY:DEGREE:TRACE 1

**:CURSOR[:TY]:DEGREE:X<x>?**

**Function** Queries the angle of the angle cursor of the T-Y display.

**Syntax** :CURSOR:TY:DEGREE:X<x>?  
<x>=1 or 2

**Example** (Example of the query of Cursor1.)  
:CURSOR:TY:DEGREE:X1? -> :CURSOR:  
TY:DEGREE:X1 -120.00000E+00

**:CURSOR[:TY]:DEGREE:Y<x>?**

**Function** Queries the voltage of the angle cursor of the T-Y display.

**Syntax** :CURSOR:TY:DEGREE:Y<x>?  
<x>=1 or 2

**Example** (Example of the query of Cursor1.)  
:CURSOR:TY:DEGREE:Y1? -> :CURSOR:  
TY:DEGREE:Y1 10.000000E-03

**:CURSOR[:TY]:HORIZONTAL?**

**Function** Queries all settings related to the H cursor of the T-Y display.

**Syntax** :CURSOR:TY:HORIZONTAL?

**Example** :CURSOR:TY:HORIZONTAL? -> :CURSOR:  
TY:HORIZONTAL:TRACE 3;  
POSITION1 2.00;POSITION2 3.00

**:CURSOR[:TY]:HORIZONTAL:DY?**

**Function** Queries the Y-axis value between H cursors of the T-Y display.

**Syntax** :CURSOR:TY:HORIZONTAL:DY?

**Example** :CURSOR:TY:HORIZONTAL:DY? ->  
:CURSOR:TY:HORIZONTAL:  
DY 3.0000000E+00

**Description** When linear scaling is ON, the query is for the scaled value.

**:CURSOR[:TY]:HORIZONTAL:POSITION<x>**

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

**Syntax** :CURSOR:TY:HORIZONTAL:POSITION<x>  
{<NRf>}  
:CURSOR:TY:HORIZONTAL:POSITION<x>?  
<x>=1 or 2  
<NRf>=5 to 5 div (1/100 steps)

**Example** :CURSOR:TY:HORIZONTAL:POSITION1 2  
:CURSOR:TY:HORIZONTAL:POSITION1? ->  
:CURSOR:TY:HORIZONTAL:  
POSITION1 2.00

**Description** <NRf> is valid down to two digits right of the decimal point.

**:CURSOR[:TY]:HORIZONTAL:TRACE**

**Function** Sets the target waveform of the H cursor of the T-Y display or queries the current setting.

**Syntax** :CURSOR:TY:HORIZONTAL:TRACE  
{<NRf>|MATH<x>}  
:CURSOR:TY:HORIZONTAL:TRACE?  
<NRf>=1 to 16  
<x>=1 to 8

**Example** :CURSOR:TY:HORIZONTAL:TRACE 1  
:CURSOR:TY:HORIZONTAL:TRACE? ->  
:CURSOR:TY:HORIZONTAL:TRACE 1

**:CURSOR[:TY]:HORIZONTAL:Y<x>?**

**Function** Queries the Y-axis value of the H cursor of the T-Y display.

**Syntax** :CURSOR:TY:HORIZONTAL:Y<x>?  
<x>=1 or 2

**Example** :CURSOR:TY:HORIZONTAL:Y1? ->  
:CURSOR:TY:HORIZONTAL:  
Y1 -1.5000000E+00

**Description** When linear scaling is ON, the query is for the scaled value.

**:CURSOR[:TY]:LOGIC?**

**Function** Queries all settings related to the logic display of the T-Y display.

**Syntax** :CURSOR:TY:LOGIC?

**Example** :CURSOR:TY:LOGIC:? -> :CURSOR:TY:  
LOGIC:FORMAT BINARY;ORDER A1\_B8

**:CURSOR[:TY]:LOGIC:FORMAT**

**Function** Sets the numerical notation of the logic display of the T-Y display.

**Syntax** :CURSOR:TY:LOGIC:FORMAT  
{BINARY|HEXA}

**Example** :CURSOR:TY:LOGIC:FORMAT BINARY  
:CURSOR:TY:LOGIC:FORMAT? ->  
:CURSOR:TY:LOGIC:FORMAT BINARY

**:CURSOR[:TY]:LOGIC:ORDER**

**Function** Sets the higher/lower bit of the logic display of the T-Y display.

**Syntax** :CURSOR:TY:LOGIC:ORDER  
{A1\_B8|B8\_A1}

**Example** :CURSOR:TY:LOGIC:ORDER A1\_B8  
:CURSOR:TY:LOGIC:ORDER? -> :CURSOR:  
TY:LOGIC:ORDER A1\_B8

**:CURSOR[:TY]:MARKER?**

**Function** Queries all settings related to the marker cursor of the T-Y display.

**Syntax** :CURSOR:TY:MARKER?

**Example** :CURSOR:TY:MARKER? -> :CURSOR:TY:  
MARKER:M1:TRACE 1;  
POSITION -4.0000000;:CURSOR:TY:  
MARKER:M2:TRACE 2;  
POSITION -1.0000000;:CURSOR:TY:  
MARKER:M3:TRACE 3;  
POSITION 1.0000000;:CURSOR:TY:  
MARKER:M4:TRACE 4;  
POSITION 3.0000000

## 6.9 CURSor Group

### :CURSor[:TY]:MARKer:DX<x>?

**Function** Queries the X-axis value between the marker cursors of the T-Y display.

**Syntax** :CURSor:TY:MARKer:DX<x>?<x>=1 to 4

**Example** :CURSOR:TY:MARKER:DX2? -> :CURSOR:TY:MARKER:DX2 600.00000E-03

**Description**

- **When the time base is set to internal clock**  
When in the time domain, the query is for the time between the markers; when in the frequency domain, the query is for the frequency between the markers.
- **When the time base is set to external clock**  
The query is for the number of points between the marker cursors.

### :CURSor[:TY]:MARKer:DY<x>?

**Function** Queries the physical value of the Y-axis between the marker cursors of the T-Y display.

**Syntax** :CURSor:TY:MARKer:DY<x>?<x>=1 to 4

**Example** :CURSOR:TY:MARKER:DY2? -> :CURSOR:TY:MARKER:DY2 666.66667E-03

**Description** When linear scaling is ON, the query is for the scaled value.

### :CURSor[:TY]:MARKer:M<x>?

**Function** Queries all settings related to the specified marker cursor.

**Syntax** :CURSor:TY:MARKer:M<x>?<x>=1 to 4

**Example** :CURSOR:TY:MARKER:M1? -> :CURSOR:TY:MARKER:M1:TRACE 1;POSITION -4.0000000

### :CURSor[:TY]:MARKer:M<x>:JUMP

**Function** Jumps the marker cursor of the T-Y display onto the specified zoom waveform.

**Syntax** :CURSor:MARKer:M<x>:JUMP {Z1|Z2}<x>=1 to 4

**Example** :CURSOR:TY:MARKER:M1:JUMP Z1

**Description** Jumps to the center position of the zoom waveform.

### :CURSor[:TY]:MARKer:M<x>:POSITION

**Function** Sets the X-axis value at the marker cursor position of the T-Y display or queries the current setting.

**Syntax** :CURSor:TY:MARKer:M<x>:POSITION {<NRF>}<x> of M<x>=1 to 4<NRF>=-5 to 5 divisions (10 divisions/displayed record length steps)

**Example** :CURSOR:TY:MARKER:M1:POSITION -1  
:CURSOR:TY:MARKER:M1:POSITION? -> :CURSOR:TY:MARKER:M1:POSITION -4.0000000

### :CURSor[:TY]:MARKer:M<x>:TRACE

**Function** Sets the target waveform of the marker cursor of the T-Y display or queries the current setting.

**Syntax** :CURSor:TY:MARKer:M<x>:TRACE {<NRF>|MATH<x>}<x> of M<x>=1 to 4<NRF>=1 to 16<x> of MATH<x>=1 to 8

**Example** :CURSOR:TY:MARKER:M1:TRACE 1  
:CURSOR:TY:MARKER:M1:TRACE? -> :CURSOR:TY:MARKER:M1:TRACE 1

### :CURSor[:TY]:MARKer:X<x>?

**Function** Queries the X-axis value at the marker cursor position of the T-Y display.

**Syntax** :CURSor:TY:MARKer:X<x>?<x>=1 to 4

**Example** :CURSOR:TY:MARKER:X1? -> :CURSOR:TY:MARKER:X1 -4.5000E-03

**Description**

- **When the time base is set to internal clock**  
When in the time domain, the query is for the time; when in the frequency domain, the query is for the frequency.
- **When the time base is set to external clock**  
When in the time domain, the query is for the number of points with respect to delay 0; when in the frequency domain, the query is for the number of points with respect to the head of the memory.

### :CURSor[:TY]:MARKer:Y<x>?

**Function** Queries the Y-axis value at the marker cursor position of the T-Y display.

**Syntax** :CURSor:TY:MARKer:Y<x>?<x>=1 to 4

**Example** :CURSOR:TY:MARKER:Y1? -> :CURSOR:TY:MARKER:Y1 1.5000E-03

**Description** When linear scaling is ON, the query is for the scaled value.

### :CURSor[:TY]:TYPE

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

**Syntax** :CURSor:TY:TYPE {OFF|HORIZONTAL|VERTICAL|MARKer|DEGREE}<x> of MATH<x>=1 to 8

**Example** :CURSOR:TY:TYPE HORIZONTAL  
:CURSOR:TY:TYPE? -> :CURSOR:TY:TYPE HORIZONTAL

### :CURSor[:TY]:VERTical?

**Function** Queries all settings related to V cursor of the T-Y display.

**Syntax** :CURSor:TY:VERTical?

**Example** :CURSOR:TY:VERTICAL? -> :CURSOR:TY:VERTICAL:TRACE 1;POSITION1 2.0000000;POSITION2 4.0000000

**:CURSOR[:TY]:VERTICAL:DX?**

**Function** Queries the X-axis value between the V cursors of the T-Y display.

**Syntax** :CURSOR:TY:VERTICAL:DX?

**Example** :CURSOR:TY:VERTICAL:DX? -> :CURSOR:TY:VERTICAL:DX 2.0000000E-03

**Description**

- **When the time base is set to internal clock**  
When in the time domain, the query is for the time between the markers; when in the frequency domain, the query is for the frequency between the markers.
- **When the time base is set to external clock**  
The query is for the number of points between the marker cursors.

**:CURSOR[:TY]:VERTICAL:DY?**

**Function** Queries the Y-axis value between V cursors of the T-Y display.

**Syntax** :CURSOR:TY:VERTICAL:DY?

**Example** :CURSOR:TY:VERTICAL:DY? -> :CURSOR:TY:VERTICAL:DY -10.0000000E-03

**Description** When linear scaling of the Y trace is ON, the query is for the scaled value.

**:CURSOR[:TY]:VERTICAL:JUMP**

**Function** Jumps the V cursor on the T-Y display onto the specified zoom waveform.

**Syntax** :CURSOR:TY:VERTICAL:JUMP  
{C1\_Z1|C1\_Z2|C2\_Z1|C2\_Z2}

**Example** :CURSOR:TY:VERTICAL:JUMP C1\_Z1

**Description** Jumps to the center position of the zoom waveform. C1 and C2 indicate V cursors Cursor1 and Cursor2, respectively.

**:CURSOR[:TY]:VERTICAL:PERDt?**

**Function** Queries the 1/|s|DT value between V cursors of the T-Y display.

**Syntax** :CURSOR:TY:VERTICAL:PERDt?

**Example** :CURSOR:TY:VERTICAL:PERDt? ->  
:CURSOR:TY:VERTICAL:  
PERDt 500.000000E+00

**Description** When the target trace is in the frequency domain, "NAN (Not A Number)" is returned.

**:CURSOR[:TY]:VERTICAL:POSITION<x>**

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

**Syntax** :CURSOR:TY:VERTICAL:POSITION<x>  
{<NRf>}  
:CURSOR:TY:VERTICAL:POSITION<x>?  
<x>=1 or 2

<NRf>=-5 to 5 divisions (10 divisions/displayed record length steps)

**Example** :CURSOR:TY:VERTICAL:POSITION1 2  
:CURSOR:TY:VERTICAL:POSITION1? ->  
:CURSOR:TY:VERTICAL:  
POSITION1 2.0000000

**:CURSOR[:TY]:VERTICAL:TRACE**

**Function** Sets the target waveform of the V cursor of the T-Y display or queries the current setting.

**Syntax** :CURSOR:TY:VERTICAL:TRACE  
{<NRf>|MATH<x>|ALL|PODA|PODB|PODA\_POdB}  
:CURSOR:TY:VERTICAL:TRACE?  
<NRf>=1 to 16  
<x>=1 to 8

**Example** :CURSOR:TY:VERTICAL:TRACE 1  
:CURSOR:TY:VERTICAL:TRACE? ->  
:CURSOR:TY:VERTICAL:TRACE 1

**:CURSOR[:TY]:VERTICAL:X<x>?**

**Function** Queries the X-axis value at the V cursor position of the T-Y display.

**Syntax** :CURSOR:TY:VERTICAL:X<x>?  
<x>=1 or 2

**Example** :CURSOR:TY:VERTICAL:X1? -> :CURSOR:TY:VERTICAL:X1 2.0000000E-03

**Description**

- **When the time base is set to internal clock**  
When in the time domain, the query is for the time; when in the frequency domain, the query is for the frequency.
- **When the time base is set to external clock**  
When in the time domain, the query is for the number of points with respect to delay 0; when in the frequency domain, the query is for the number of points with respect to the head of the memory.

**:CURSOR[:TY]:VERTICAL:Y<x>?**

**Function** Queries the Y-axis value at the V cursor position of the T-Y display.

**Syntax** :CURSOR:TY:VERTICAL:Y<x>?  
<x>=1 or 2

**Example** :CURSOR:TY:VERTICAL:Y1? -> :CURSOR:TY:VERTICAL:Y1 0.0000000E+00

**:CURSOR:XY?**

**Function** Queries all settings related to the cursor of the X-Y display.

**Syntax** :CURSOR:XY?

**Example** :CURSOR:XY? -> :CURSOR:XY:  
TYPE VERTICAL;HORIZONTAL:TRACE XY1;  
POSITION1 5.00;POSITION2 -5.00;  
STRACE 3;;CURSOR:XY:VERTICAL:  
TRACE XY1;POSITION1 2.00;  
POSITION2 5.00;STRACE 1;;CURSOR:XY:  
MARKER:TRACE XY1;  
POSITION -1.0000000;STRACE 2

**:CURSOR:XY:HORIZONTAL?**

**Function** Queries all settings related to the H cursor of the X-Y display.

**Syntax** :CURSOR:XY:HORIZONTAL?

**Example** :CURSOR:XY:HORIZONTAL? -> :CURSOR:XY:HORIZONTAL:TRACE XY1;  
POSITION1 5.00;POSITION2 -5.00;  
STRACE 3

## 6.9 CURSOR Group

### **:CURSOR:XY:HORIZONTAL:DY?**

**Function** Queries the Y-axis value between H cursors of the X-Y display.

**Syntax** :CURSOR:XY:HORIZONTAL:DY?

**Example** :CURSOR:XY:HORIZONTAL:DY? ->  
:CURSOR:XY:HORIZONTAL:  
DY 500.00000E+00

**Description** When linear scaling is ON, the query is for the scaled value.

### **:CURSOR:XY:HORIZONTAL:POSITION<x>**

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

**Syntax** :CURSOR:XY:HORIZONTAL:POSITION<x>  
{<NRF>}  
:CURSOR:XY:HORIZONTAL:POSITION<x>?  
<x>=1 or 2

**Example** :CURSOR:XY:HORIZONTAL:POSITION1 2  
:CURSOR:XY:HORIZONTAL:POSITION1? ->  
:CURSOR:XY:HORIZONTAL:  
POSITION1 2.00

### **:CURSOR:XY:HORIZONTAL:TRACE**

**Function** Sets the target waveform of the H cursor of the X-Y display or queries the current setting.

**Syntax** :CURSOR:XY:HORIZONTAL:TRACE  
{XY1|XY2|XY3|XY4}  
:CURSOR:XY:HORIZONTAL:TRACE?

**Example** :CURSOR:XY:HORIZONTAL:TRACE XY1  
:CURSOR:XY:HORIZONTAL:TRACE? ->  
:CURSOR:XY:HORIZONTAL:TRACE XY1

### **:CURSOR:XY:HORIZONTAL:STRACE**

**Function** Sets the target waveform of the H cursor of the X-Y display when in Single mode or queries the current setting.

**Syntax** :CURSOR:XY:HORIZONTAL:STRACE  
{<NRF>|MATH<x>}  
:CURSOR:XY:HORIZONTAL:STRACE?  
<NRF>=1 to 16  
<x>=1 to 8

**Example** :CURSOR:XY:HORIZONTAL:STRACE 3  
:CURSOR:XY:HORIZONTAL:STRACE? ->  
:CURSOR:XY:HORIZONTAL:STRACE 3

### **:CURSOR:XY:HORIZONTAL:Y<x>?**

**Function** Queries the Y-axis value of the H cursor of the X-Y display.

**Syntax** :CURSOR:XY:HORIZONTAL:Y<x>?  
<x>=1 or 2

**Example** :CURSOR:XY:HORIZONTAL:Y1? ->  
:CURSOR:XY:HORIZONTAL:  
Y1 250.00000E+00

**Description** When linear scaling is ON, the query is for the scaled value.

### **:CURSOR:XY:MARKER?**

**Function** Queries all settings related to the marker cursor of the X-Y display.

**Syntax** :CURSOR:XY:MARKER?

**Example** :CURSOR:XY:MARKER? -> :CURSOR:XY:  
MARKER:TRACE XY1;  
POSITION -1.0000000;STRACE 2

### **:CURSOR:XY:MARKER:POSITION**

**Function** Sets the X-axis value at the marker cursor position of the X-Y display or queries the current setting.

**Syntax** :CURSOR:XY:MARKER:POSITION {<NRF>}  
:CURSOR:XY:MARKER:POSITION?  
<NRF>=-5 to 5 divisions (10 divisions/displayed  
record length steps)

**Example** :CURSOR:XY:MARKER:POSITION -1  
:CURSOR:XY:MARKER:POSITION? ->  
:CURSOR:XY:MARKER:  
POSITION -1.0000000

### **:CURSOR:XY:MARKER:T?**

**Function** Queries the time value of the marker cursor of the X-Y display.

**Syntax** :CURSOR:XY:MARKER:T?

**Example** :CURSOR:XY:MARKER:T? -> :CURSOR:XY:  
MARKER:T -1.0000000E-03

### **:CURSOR:XY:MARKER:TRACE**

**Function** Sets the target waveform of the marker cursor of the X-Y display when the X-axis setting is Quad or queries the current setting.

**Syntax** :CURSOR:XY:MARKER:TRACE  
{XY1|XY2|XY3|XY4}  
:CURSOR:XY:MARKER:TRACE?

**Example** :CURSOR:XY:MARKER:TRACE XY1  
:CURSOR:XY:MARKER:TRACE? ->  
:CURSOR:XY:MARKER:TRACE XY1

### **:CURSOR:XY:MARKER:STRACE**

**Function** Sets the target waveform of the marker cursor of the X-Y display when the X-axis setting is Single or queries the current setting.

**Syntax** :CURSOR:XY:MARKER:STRACE  
{<NRF>|MATH<x>}  
:CURSOR:XY:MARKER:STRACE?  
<NRF>=1 to 16  
<x>=1 to 8

**Example** :CURSOR:XY:MARKER:STRACE 2  
:CURSOR:XY:MARKER:STRACE? ->  
:CURSOR:XY:MARKER:STRACE 2

### **:CURSOR:XY:MARKER:X?**

**Function** Queries the X-axis value at the marker cursor position of the X-Y display.

**Syntax** :CURSOR:XY:MARKER:X?

**Example** :CURSOR:XY:MARKER:X? -> :CURSOR:  
XY:MARKER:X 3.6875000E+00

**Description** When linear scaling is ON, the query is for the scaled value.

### **:CURSOR:XY:MARKER:Y?**

**Function** Queries the Y-axis value at the marker cursor position of the X-Y display.

**Syntax** :CURSOR:XY:MARKER:Y?

**Example** :CURSOR:XY:MARKER:Y? -> :CURSOR:XY:  
MARKER:Y -166.66667E-03

**Description** When linear scaling is ON, the query is for the scaled value.

**:CURSOR:XY:TYPE**

Function Sets the cursor type of the X-Y display or queries the current setting.

Syntax :CURSOR:XY:TYPE {OFF|HORIZONTAL|VERTICAL|MARKER|HAVERTICAL}  
:CURSOR:XY:TYPE?

Example :CURSOR:XY:TYPE VERTICAL  
:CURSOR:XY:TYPE? -> :CURSOR:XY:TYPE VERTICAL

**:CURSOR:XY:VERTICAL?**

Function Queries all settings related to the V cursor of the X-Y display.

Syntax :CURSOR:XY:VERTICAL?

Example :CURSOR:XY:VERTICAL? -> :CURSOR:XY:VERTICAL:TRACE XY1;POSITION1 2.00;POSITION2 5.00;STRACE 1

**:CURSOR:XY:VERTICAL:DX?**

Function Queries the X-axis value between the V cursors of the X-Y display.

Syntax :CURSOR:XY:VERTICAL:DX?

Example :CURSOR:XY:VERTICAL:DX? -> :CURSOR:XY:VERTICAL:DX 30.000000E+00

Description When linear scaling of the X trace is ON, the query is for the scaled value.

**:CURSOR:XY:VERTICAL:POSITION<x>**

Function Sets the V cursor position of the X-Y display or queries the current setting.

Syntax :CURSOR:XY:VERTICAL:POSITION<x>  
{<NRf>}  
:CURSOR:XY:VERTICAL:POSITION<x>?  
<x>=1 or 2  
<NRf>=-5 to 5 div (1/100 steps)

Example :CURSOR:XY:VERTICAL:POSITION1 2  
:CURSOR:XY:VERTICAL:POSITION1? ->  
:CURSOR:XY:VERTICAL:POSITION1 2.00

**:CURSOR:XY:VERTICAL:TRACE**

Function Sets the target waveform of the V cursor of the X-Y display when the X-axis setting is Quad or queries the current setting.

Syntax :CURSOR:XY:VERTICAL:TRACE  
{XY1|XY2|XY3|XY4}  
:CURSOR:XY:VERTICAL:TRACE?

Example :CURSOR:XY:VERTICAL:TRACE XY1  
:CURSOR:XY:VERTICAL:TRACE? ->  
:CURSOR:XY:VERTICAL:TRACE XY1

**:CURSOR:XY:VERTICAL:STRACE?**

Function Queries the target waveform of the V cursor of the X-Y display when the X-axis setting is Single.

Syntax :CURSOR:XY:VERTICAL:STRACE?

Example :CURSOR:XY:VERTICAL:STRACE? ->  
:CURSOR:XY:VERTICAL:STRACE 2

**:CURSOR:XY:VERTICAL:X<x>?**

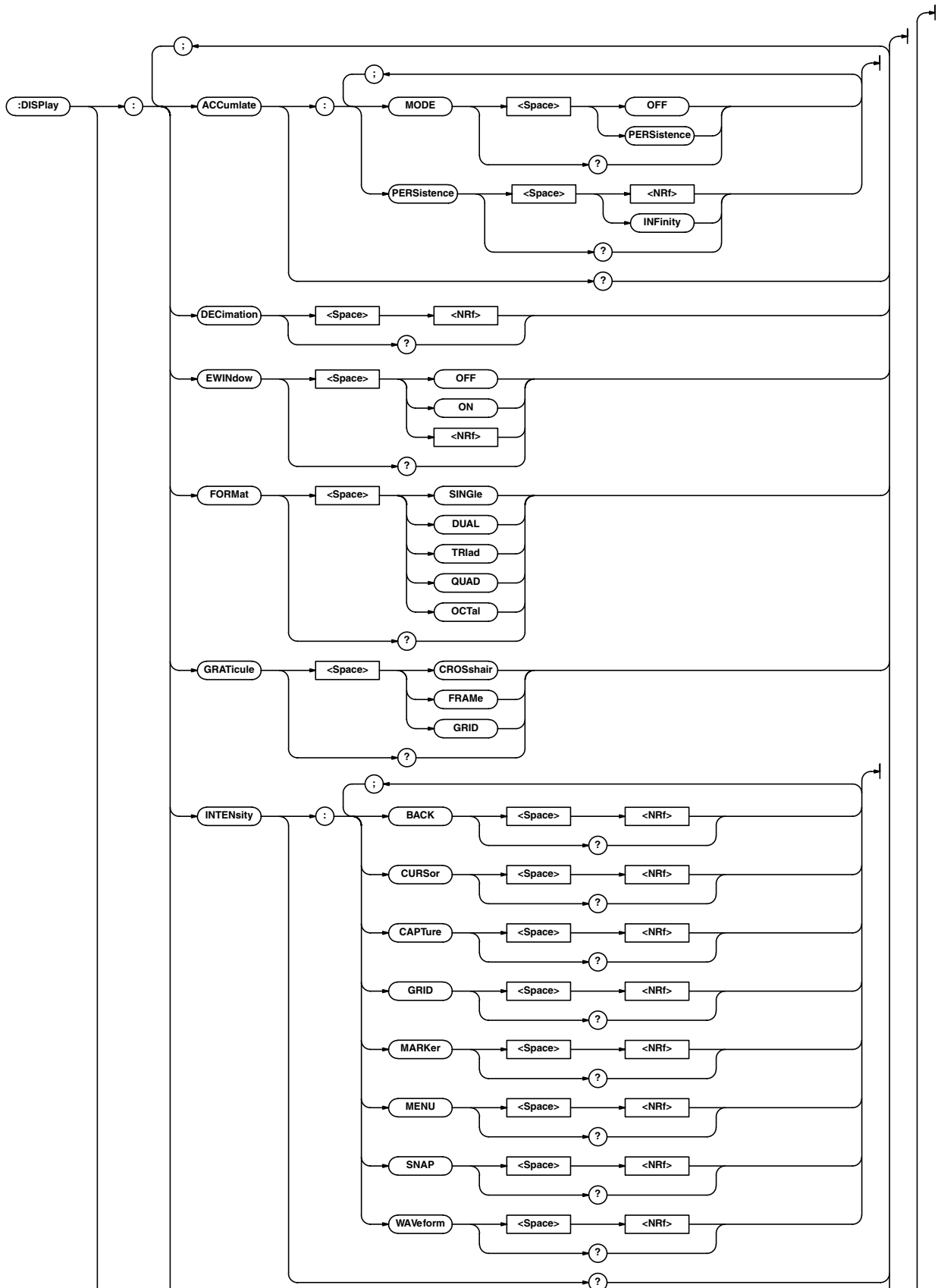
Function Queries the X-axis value at the V cursor position of the X-Y display.

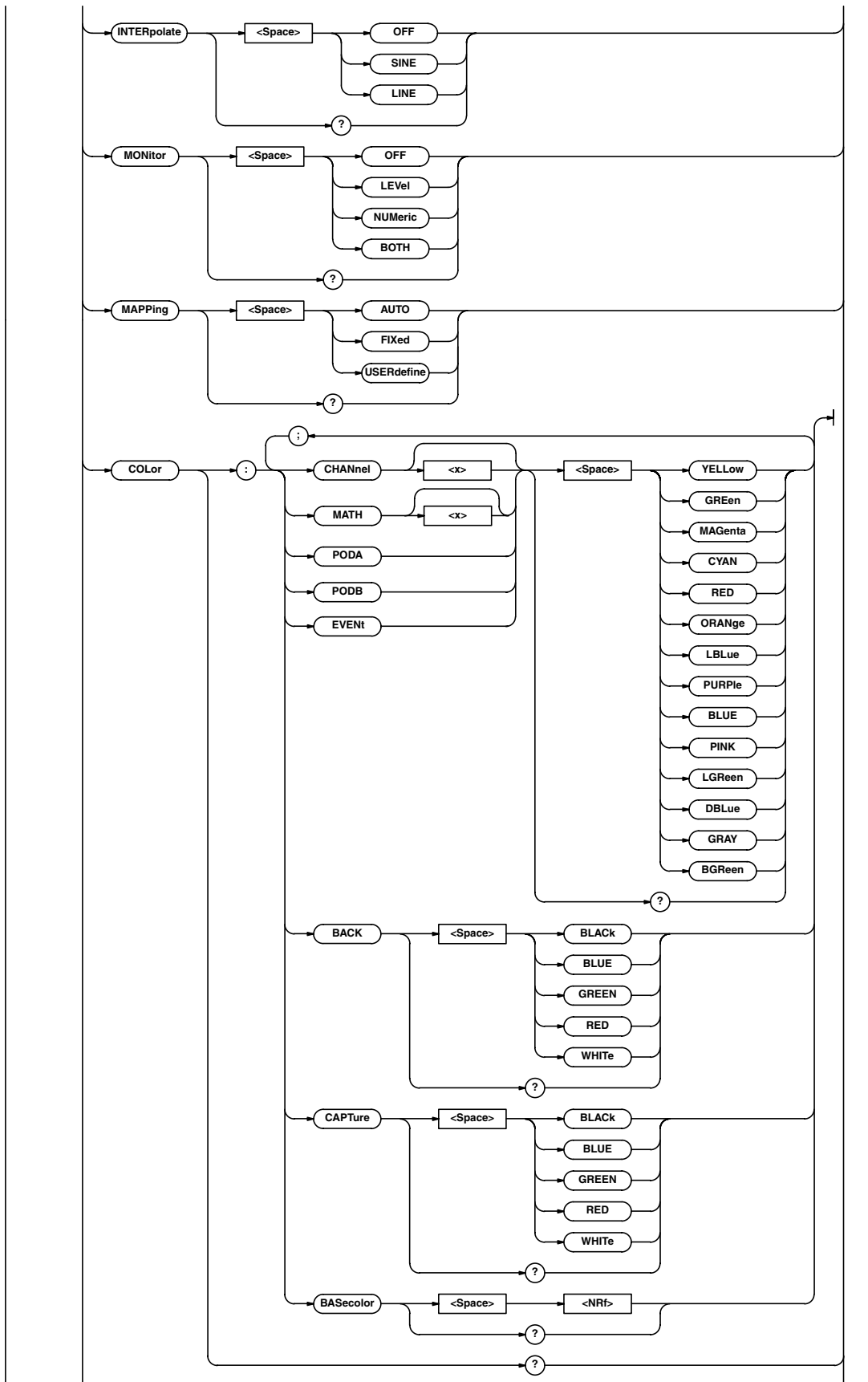
Syntax :CURSOR:XY:VERTICAL:X<x>?  
<x>=1 or 2

Example :CURSOR:XY:VERTICAL:X1? ->  
:CURSOR:XY:VERTICAL:X1 20.000000E+00

### 6.10 DISPlay Group

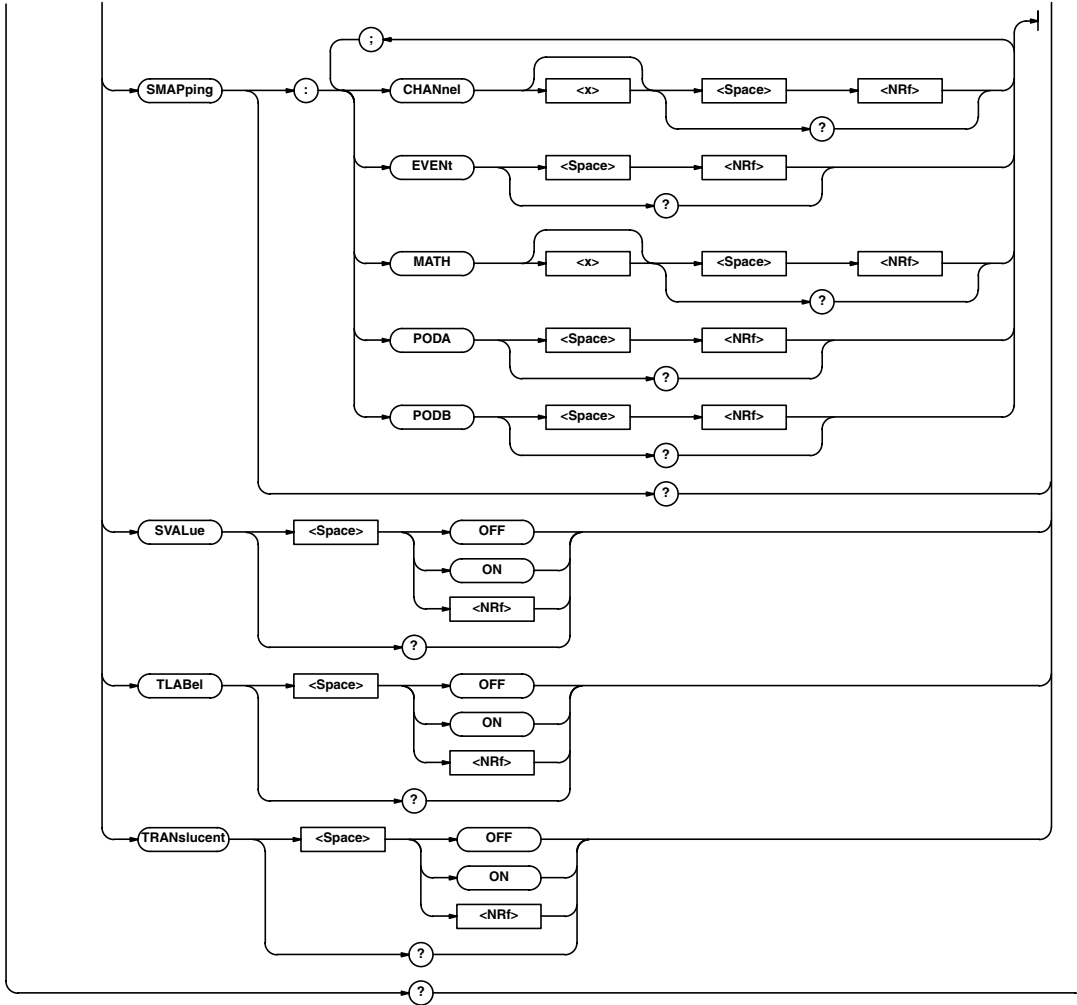
The commands in this group deal with the screen display. You can make the same settings and inquiries as when the front panel keys such as the DISPLAY key and MISC key are used.







## 6.10 DISPlay Group



### :DISPlay?

Function Queries all settings related to the display.

Syntax :DISPlay?

Example :DISPLAY? -> :DISPLAY:FORMAT QUAD;  
 INTERPOLATE LINE;DECIMATION 2000;  
 GRATICULE GRID;ACCUMLATE:MODE OFF;  
 PERSISTENCE IN FINITY;:DISPLAY:  
 EWINDOW 0;SVALUE 1;TLABEL 0;  
 MONITOR BOTH;TRANSLUCENT 0;  
 MAPPING AUTO;SMAPPING:CHANNEL1 0;  
 CHANNEL2 1;CHANNEL3 2;CHANNEL4 3;  
 CHANNEL5 4;CHANNEL6 5;CHANNEL7 6;  
 CHANNEL8 7;CHANNEL9 0;CHANNEL10 1;  
 CHANNEL11 2;CHANNEL12 3;  
 CHANNEL13 4;CHANNEL14 5;  
 CHANNEL15 6;CHANNEL16 7;MATH1 6;  
 MATH2 7;MATH3 0;MATH4 1;MATH5 2;  
 MATH6 3;MATH7 4;MATH8 5;PODA 6;  
 PODB 7;EVENT 0;:DISPLAY:COLOR:  
 CHANNEL1 YELLOW;CHANNEL2 GREEN;  
 CHANNEL3 MAGENTA;CHANNEL4 CYAN;  
 CHANNEL5 RED;CHANNEL6 ORANGE;  
 CHANNEL7 LBLUE;CHANNEL8 PURPLE;  
 CHANNEL9 PINK;CHANNEL10 LGREEN;  
 CHANNEL11 DBLUE;CHANNEL12 YELLOW;  
 CHANNEL13 GREEN;CHANNEL14 MAGENTA;  
 CHANNEL15 CYAN;CHANNEL16 RED;  
 MATH1 YELLOW;MATH2 GREEN;  
 MATH3 MAGENTA;MATH4 CYAN;MATH5 RED;

MATH6 ORANGE;MATH7 LBLUE;  
 MATH8 PURPLE;PODA PINK;PODB LGREEN;  
 BACK BLACK;CAPTURE BLACK;  
 BASECOLOR 1;:DISPLAY:INTENSITY:  
 WAVEFORM 12;MENU 12;BACK 3;  
 CAPTURE 3;SNAP 12;GRID 12;  
 CURSOR 12;MARKER 12

### :DISPlay:ACCumulate?

Function Queries all settings related to the waveform accumulation display.

Syntax :DISPlay:ACCumulate?

Example :DISPLAY:ACCUMULATE? -> :DISPLAY:  
 ACCUMULATE:MODE PERSISTENCE;  
 PERSISTENCE 16

### :DISPlay:ACCumulate:MODE

Function Sets the waveform accumulation display mode or queries the current setting.

Syntax :DISPlay:ACCumulate:MODE  
 {OFF|PERSISTENCE}  
 :DISPlay:ACCumulate:MODE?

Example :DISPLAY:ACCUMULATE:  
 MODE PERSISTENCE  
 :DISPLAY:ACCUMULATE:MODE? ->  
 :DISPLAY:ACCUMULATE:  
 MODE PERSISTENCE

**:DISPlay:ACCumulate:PERSistence**

Function Sets the number of accumulated waveforms or queries the current setting.

Syntax :DISPlay:ACCumulate:PERSistence  
{<NRf>|INfInity}  
:DISPlay:ACCumulate:PERSistence?  
<NRf>=2 to 128 (2<sup>n</sup> steps)

Example :DISPLAY:ACCUMULATE:PERSISTENCE 16  
:DISPLAY:ACCUMULATE:PERSISTENCE? ->  
:DISPLAY:ACCUMULATE:PERSISTENCE 16

**:DISPlay:COLor?**

Function Queries all settings related to the display color.

Syntax :DISPlay:COLor?

Example :DISPLAY:COLOR? -> :DISPLAY:COLOR:  
CHANNEL1 YELLOW;CHANNEL2 GREEN;  
CHANNEL3 MAGENTA;CHANNEL4 CYAN;  
CHANNEL5 RED;CHANNEL6 ORANGE;  
CHANNEL7 LBLUE;CHANNEL8 PURPLE;  
CHANNEL9 PINK;CHANNEL10 LGREEN;  
CHANNEL11 DBLUE;CHANNEL12 YELLOW;  
CHANNEL13 GREEN;CHANNEL14 MAGENTA;  
CHANNEL15 CYAN;CHANNEL16 RED;  
MATH1 YELLOW;MATH2 GREEN;  
MATH3 MAGENTA;MATH4 CYAN;MATH5 RED;  
MATH6 ORANGE;MATH7 LBLUE;  
MATH8 PURPLE;PODA PINK;PODB LGREEN;  
BACK BLACK;CAPTURE BLACK;  
BASECOLOR 1

**:DISPlay:COLor:{CHANnel<x>|MATH<x>|PodA|PodB|EVENT}**

Function Sets the color of the waveform or queries the current setting.

Syntax :DISPlay:COLor:{CHANnel<x>|MATH<x>|PodA|PodB|EVENT}  
{YELLow|GREen|MAGenta|CYAN|RED|ORAnge|LBLue|PURPle|BLUE|PINK|LGRen|DBLue|GRAY|BGRen}  
:DISPlay:COLor:COLor:{CHANnel<x>|MATH<x>|PodA|PodB|EVENT}?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8

Example :DISPLAY:COLOR:CHANNEL1 MAGENTA  
:DISPLAY:COLOR:CHANNEL1? ->  
:DISPLAY:COLOR:CHANNEL1 MAGENTA

**:DISPlay:COLor:{BACK|CAPTure}**

Function Sets the color of the background or the sub waveform window or queries the current setting.

Syntax :DISPlay:COLor:{BACK|CAPTure}  
{BLACK|BLUE|RED|GREen|WHITE}  
:DISPlay:COLor:{BACK|CAPTure}?

Example :DISPLAY:COLOR:BACK WHITE  
:DISPLAY:COLOR:BACK? -> :DISPLAY:  
COLOR:BACK WHITE

**:DISPlay:COLor:BASEcolor**

Function Sets the basic color of the menu or queries the current setting.

Syntax :DISPlay:COLor:BASEcolor {<NRf>}  
:DISPlay:COLor:BASEcolor?  
<NRf>=1 to 3

Example :DISPLAY:COLOR:BASECOLOR 2  
:DISPLAY:COLOR:BASECOLOR? ->  
:DISPLAY:COLOR:BASECOLOR 2

**:DISPlay:DECimation**

Function Sets the number of decimation points or queries the current setting.

Syntax :DISPlay:DECimation {<NRf>}  
:DISPlay:DECimation?  
<NRf>=2000 or 100000

Example :DISPLAY:DECIMATION 2000  
:DISPLAY:DECIMATION? -> :DISPLAY:  
DECIMATION 2000

**:DISPlay:EWINDow**

Function Turns ON/OFF the extra window or queries the current setting.

Syntax :DISPlay:EWINDow {<Boolean>}  
:DISPlay:EWINDow?

Example :DISPLAY:EWINDOW OFF  
:DISPLAY:EWINDOW? -> :DISPLAY:  
EWINDOW 0

**:DISPlay:FORMat**

Function Sets the display format or queries the current setting.

Syntax :DISPlay:FORMat  
{SINGle|DUAL|TRIAd|QUAD|HEXa|OCTal}  
:DISPlay:FORMat?

Example :DISPLAY:FORMAT QUAD  
:DISPLAY:FORMAT? -> :DISPLAY:  
FORMAT QUAD

**:DISPlay:GRATicule**

Function Sets the grid (graticule) or queries the current setting.

Syntax :DISPlay:GRATicule  
{CROSShair|FRAME|GRID}  
:DISPlay:GRATicule?

Example :DISPLAY:GRATICULE GRID  
:DISPLAY:GRATICULE? -> :DISPLAY:  
GRATICULE GRID

**:DISPlay:INTENSity?**

Function Queries all settings related to the brightness of the displayed items.

Syntax :DISPlay:INTENSity?

Example :DISPLAY:INTENSITY? -> :DISPLAY:  
INTENSITY:WAVEFORM 12;MENU 12;  
BACK 3;CAPTURE 3;SNAP 12;GRID 12;  
CURSOR 12;MARKER 12

**:DISPlay:INTENSity:{CURSor|MENU|GRID|SNAP|WAVEform|MARKer|CAPTure|BACK}**

Function Sets the brightness of the display item or queries the current setting.

Syntax :DISPlay:INTENSity:{CURSor|MENU|GRID|SNAP|WAVEform|MARKer|CAPTure|BACK} {<NRf>}  
:DISPlay:INTENSity:{CURSor|MENU|GRID|SNAP|WAVEform|MARKer|CAPTure|BACK}?  
<NRf>=1 to 15

Example :DISPLAY:INTENSITY:CURSOR 6  
:DISPLAY:INTENSITY:CURSOR? ->  
:DISPLAY:INTENSITY:CURSOR 6

## 6.10 DISPLAY Group

### **:DISPlay:INTerpolate**

Function Sets the interpolation method or queries the current setting.

Syntax `:DISPlay:INTerpolate  
{LINE|OFF|SINE}  
:DISPlay:INTerpolate?`

Example `:DISPlay:INTerpolate LINE  
:DISPlay:INTerpolate? -> :DISPlay:  
INTerpolate LINE`

### **:DISPlay:MONitor**

Function Sets the monitor display or queries the current setting.

Syntax `:DISPlay:MONitor  
{BOTH|LEVel|OFF|NUMeric}  
:DISPlay:MONitor?`

Example `:DISPlay:MONitor BOTH  
:DISPlay:MONitor? -> :DISPlay:  
MONitor BOTH`

### **:DISPlay:MAPPing**

Function Sets the mode of the waveform mapping to the split screen or queries the current setting.

Syntax `:DISPlay:MAPPing  
{AUTO|FIXed|USERdefine}  
:DISPlay:MAPPing?`

Example `:DISPlay:MAPPing AUTO  
:DISPlay:MAPPing? -> :DISPlay:  
MAPPing AUTO`

### **:DISPlay:SMAPPing?**

Function Queries all the settings related to the waveform mapping to the split screen.

Syntax `:DISPlay:SMAPPing?`

Example `:DISPlay:SMAPPing? -> :DISPlay:  
SMAPPing:CHANnel1 3;CHANnel2 1;  
CHANnel3 2;CHANnel4 3;CHANnel5 4;  
CHANnel6 5;CHANnel7 6;CHANnel8 7;  
CHANnel9 0;CHANnel10 1;CHANnel11 2;  
CHANnel12 3;CHANnel13 4;  
CHANnel14 5;CHANnel15 6;  
CHANnel16 7;MATH1 6;MATH2 7;  
MATH3 0;MATH4 1;MATH5 2;MATH6 3;  
MATH7 4;MATH8 5;PODA 6;PODB 7;  
EVENT 0`

### **:DISPlay:SMAPPing:{CHANnel<x>| MATH<x>|PODA|PODB|EVENT}**

Function Sets the waveform mapping to the split screen or queries the current setting.

Syntax `:DISPlay:SMAPPing:{CHANnel<x>|  
MATH<x>|PODA|PODB|EVENT} {<NRf>}  
:DISPlay:SMAPPing:{CHANnel<x>|  
MATH<x>|PODA|PODB|EVENT}?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
<NRf>=0 to 7`

Example `:DISPlay:SMAPPing:CHANnel1 0  
:DISPlay:SMAPPing:CHANnel1? ->  
:DISPlay:SMAPPing:CHANnel1 3`

Description This setting is valid only when `:MATH:MAPPing` is set to `USERdefine`.

### **:DISPlay:SVALue (Scale VALUE)**

Function Turns ON/OFF the scale value display or queries the current setting.

Syntax `:DISPlay:SVALue {<Boolean>}  
:DISPlay:SVALue?`

Example `:DISPlay:SVALue OFF  
:DISPlay:SVALue? -> :DISPlay:  
SVALue 0`

### **:DISPlay:TLabel (Trace LABEL)**

Function Turns ON/OFF the waveform label display or queries the current setting.

Syntax `:DISPlay:TLabel {<Boolean>}  
:DISPlay:TLabel?`

Example `:DISPlay:TLabel ON  
:DISPlay:TLabel? -> :DISPlay:  
TLabel 1`

Description You can set the user definition of the waveform label using `:CHANnel<x>:Label`.

### **:DISPlay:TRANslucent**

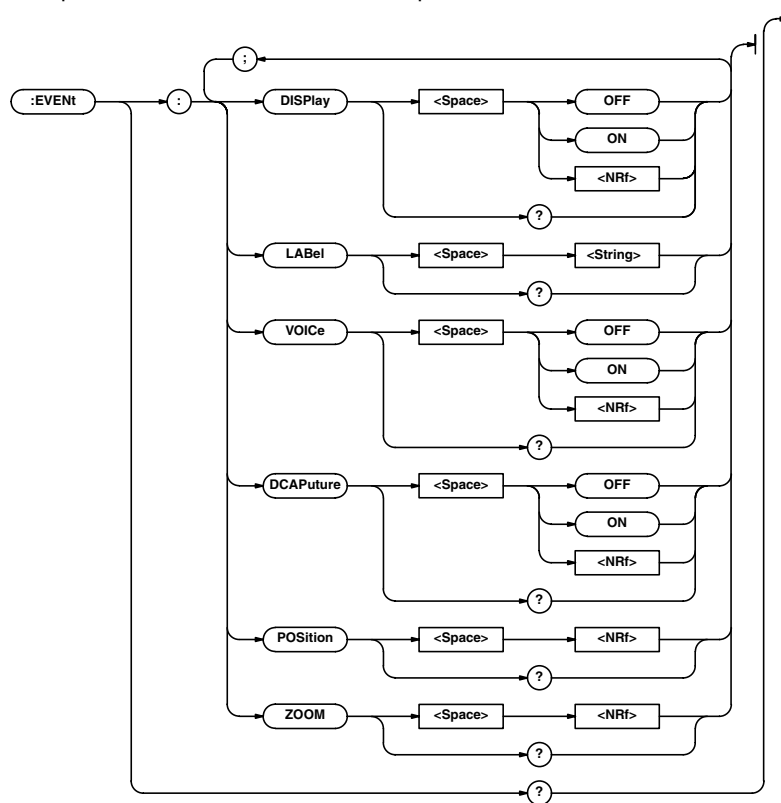
Function Turns ON/OFF the translucent mode of the pop-up menu or queries the current setting.

Syntax `:DISPlay:TRANslucent {<Boolean>}  
:DISPlay:TRANslucent?`

Example `:DISPlay:TRANslucent ON  
:DISPlay:TRANslucent? -> :DISPlay:  
TRANslucent 1`

## 6.11 EVENT Group

The commands in this group deal with event waveforms. You can make the same settings, execute the same operations, and make the same inquiries as when the SHIFT+CH11 key on the front panel is used.



### :EVENT?

**Function** Queries all settings related to the event waveform.

**Syntax** :EVENT?

**Example** :EVENT? -> :EVENT:DISPLAY 1;  
POSITION 1.00;ZOOM 0.500;VOICE 1;  
DCAPTURE 1;LABEL "Event"

### :EVENT:DISPlay

**Function** Turns ON/OFF the event waveform display or queries the current setting.

**Syntax** :EVENT:DISPlay {<Boolean>}

**Example** :EVENT:DISPLAY ON  
:EVENT:DISPLAY? -> :EVENT:DISPLAY 1

### :EVENT:LABel

**Function** Sets the event waveform label or queries the current setting.

**Syntax** :EVENT:LABel {<String>}  
<String>=Up to 8 characters

**Example** :EVENT:LABEL "ABC"  
:EVENT:LABEL? -> :EVENT:LABEL "ABC"

### :EVENT:VOICe

**Function** Sets the voice event of the event waveform or queries the current setting.

**Syntax** :EVENT:VOICe {<Boolean>}

**Example** :EVENT:VOICE ON  
:EVENT:VOICE? -> :EVENT:VOICE 1

**Description** Cannot be used on the current DL750 that you are using.

### :EVENT:DCAPture

**Function** Sets the dual capture event of the event waveform or queries the current setting.

**Syntax** :EVENT:DCAPture {<Boolean>}

**Example** :EVENT:DCAPTURE ON  
:EVENT:DCAPTURE? -> :EVENT:  
DCAPTURE 1

### :EVENT:POSition

**Function** Sets the vertical position of the event waveform or queries the current setting.

**Syntax** :EVENT:POSition {<NRf>}

<NRf>=-5.00 to 5.00 (div, 0.01 div steps)  
**Example** :EVENT:POSITION 1.0  
:EVENT:POSITION? -> :EVENT:  
POSITION 1.00

### :EVENT:ZOOM

**Function** Sets the vertical zoom of the event waveform or queries the current setting.

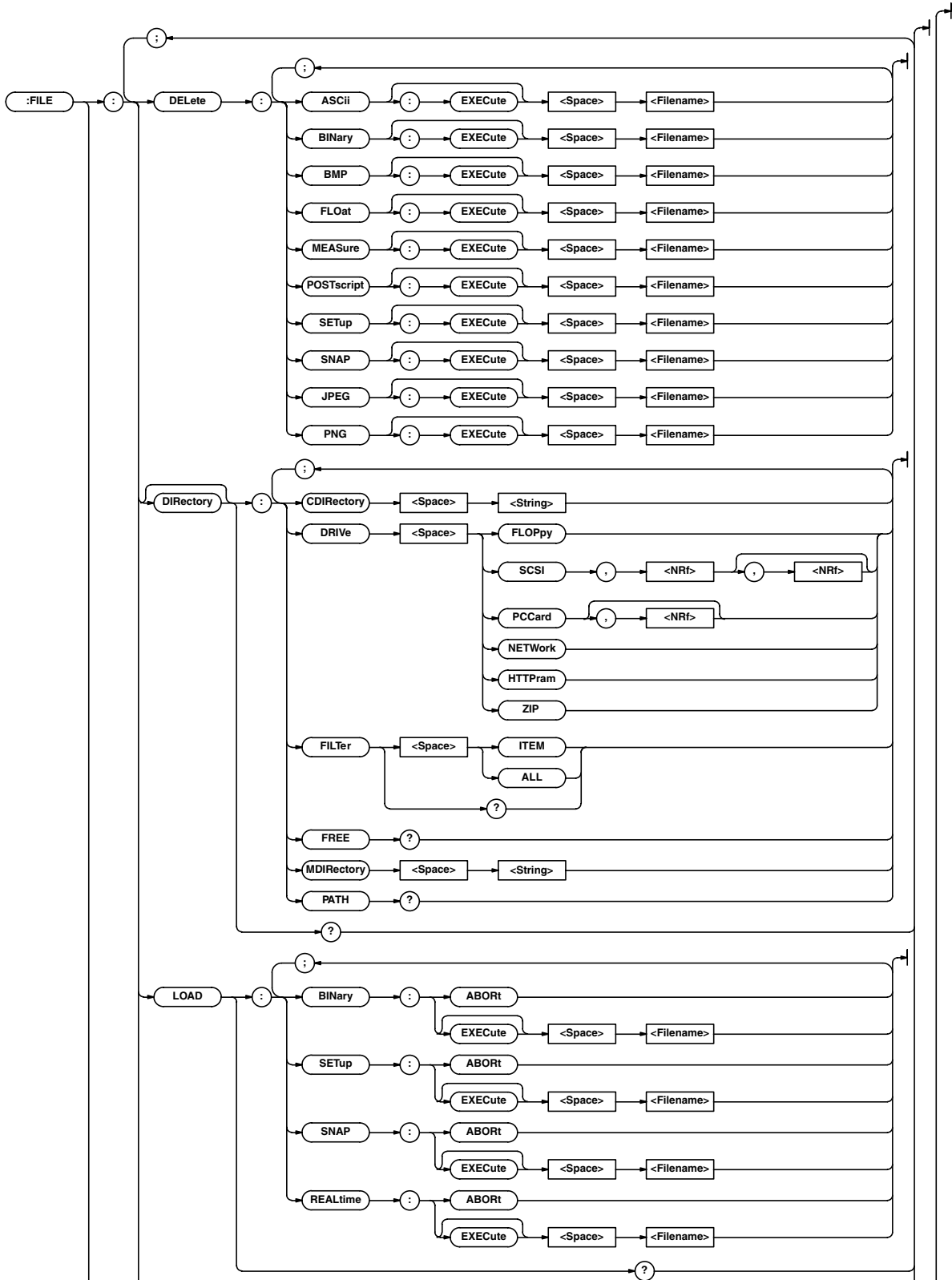
**Syntax** :EVENT:ZOOM {<NRf>}

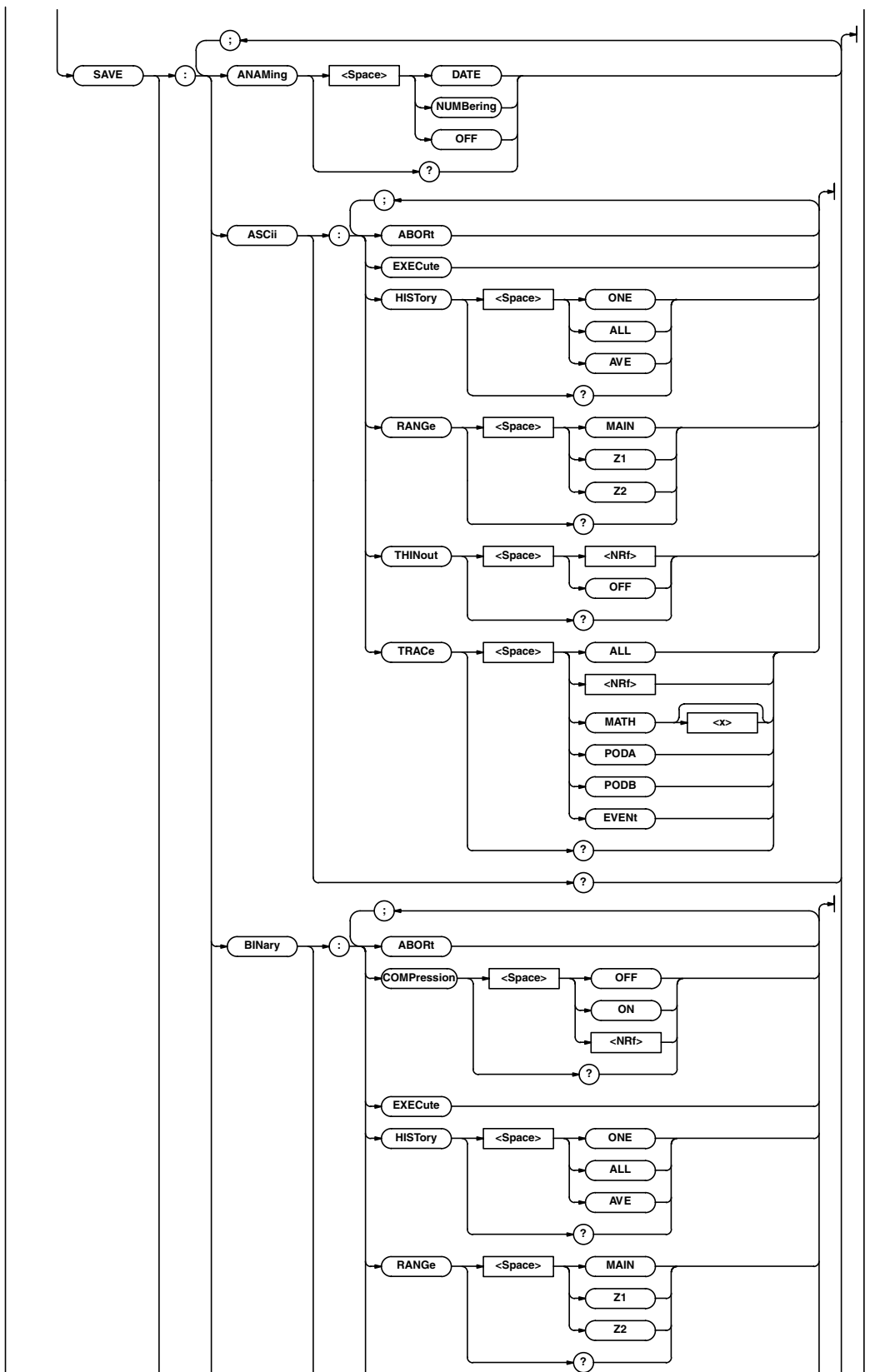
<NRf>=0.1, 0.111, 0.125, 0.143, 0.167, 0.2,  
0.25, 0.33, 0.4, 0.5, 0.556, 0.625, 0.667,  
0.714, 0.8, 0.833, 1, 1.11, 1.25, 1.33, 1.43,  
1.67, 2, 2.22, 2.5, 3.33, 4, 5, 6.67, 8, 10,  
12.5, 16.7, 20, 25, 40, 50, or 100

**Example** :EVENT:ZOOM 0.5  
:EVENT:ZOOM? -> :EVENT:ZOOM 0.500

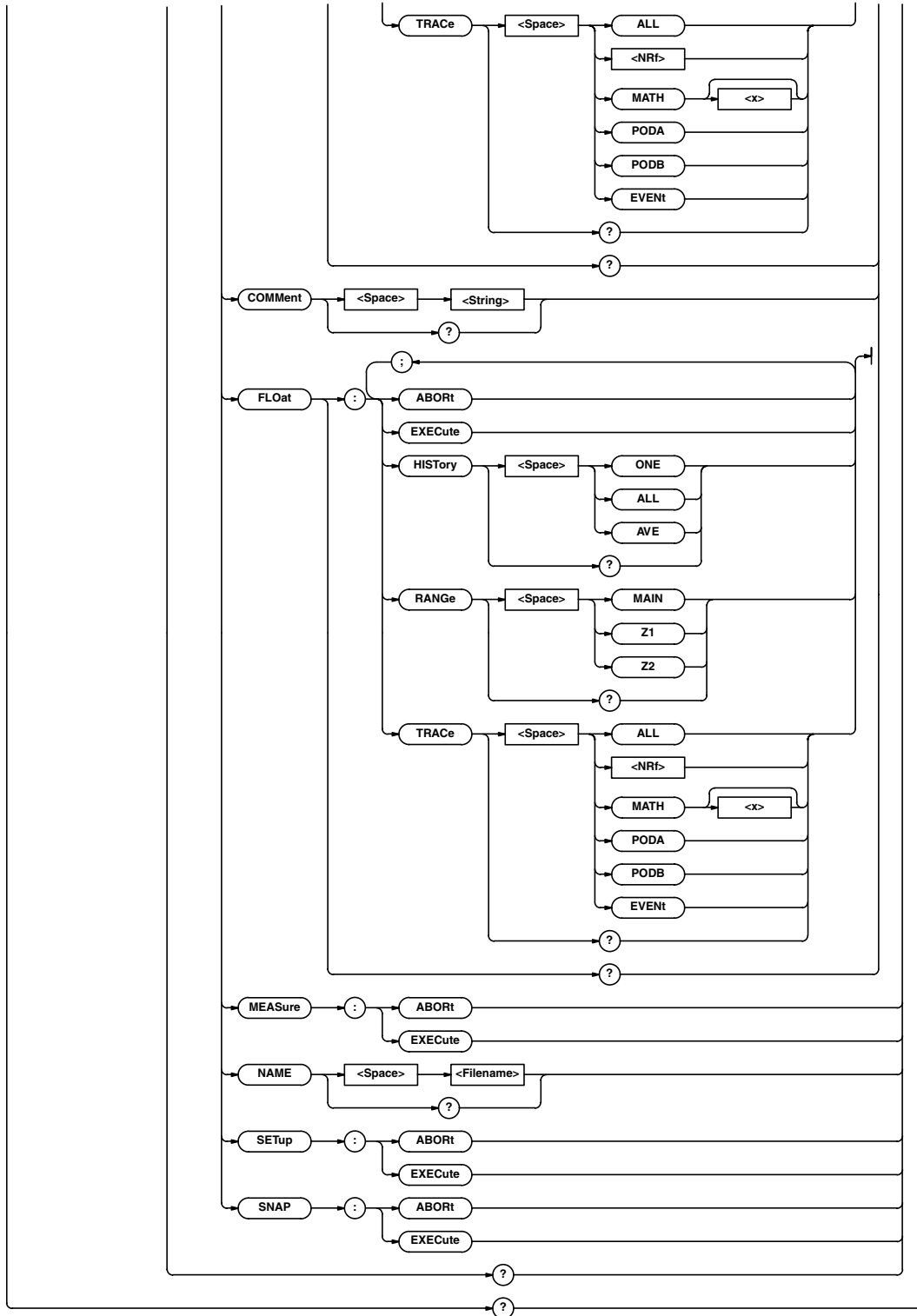
### 6.12 FILE Group

The commands in this group deal with the floppy disk, the internal hard disk, and the external SCSI device. You can make the same settings, execute the same operations, and make the same inquiries as when the FILE key on the front panel is used.





6.12 FILE Group



**:FILE?**

**Function** Queries all settings related to the specified storage medium.

**Syntax** :FILE?

**Example** :FILE? -> :FILE:DIRECTORY:  
FILTER ITEM;:FILE:SAVE:BINARy:  
TRACE ALL;RANGE MAIN;HISTORY ONE;  
COMPRESSION 0;:FILE:SAVE:ASCIi:  
TRACE ALL;RANGE MAIN;HISTORY ONE;  
INTERVAL OFF;:FILE:SAVE:FLOAT:  
TRACE ALL;RANGE MAIN;HISTORY ONE;:  
FILE:SAVE:ANAMING NUMBERING;  
NAME "";COMMENT "";:FILE:FORMAT:  
TYPE HD14;PARTITION 1

**:FILE:DELEte:{ASCIi|BINARy|BMP|  
FLOat|MEASure|POSTscript|REALtime|  
SETup|SNAP|JPEG|PNG}[:EXECute]**

**Function** Executes the deletion of the data file. This is an overlap command.

**Syntax** :FILE:DELEte:{ASCIi|BINARy|BMP|  
FLOat|MEASure|POSTscript|REALtime|  
SETup|SNAP|JPEG|PNG}[:EXECute]  
<Filename>

**Example** (The following is an example for the setup data.)  
:FILE:DELEte:SETup "CASE1"

**Description**

- The target medium to be deleted is selected using :FILE[:DIRectory]:DRIVE.
- This command cannot be executed on data on the REALTIME partition (SCSI address=internal ID, partition=0).

**:FILE:DIRectory?**

**Function** Queries all settings related to the current storage medium.

**Syntax** :FILE:DIRectory?

**Example** :FILE:DIRectory? -> :FILE:  
DIRectory:FILTEr ITEM

**:FILE[:DIRectory]:CDIRectory (Change  
Directory)**

**Function** Changes the current directory.

**Syntax** :FILE[:DIRectory]:CDIRectory  
<String>

**Example** :FILE:DIRectory:CDIRectory "NO\_1"

**:FILE[:DIRectory]:DRIVE**

**Function** Sets the current storage medium.

**Syntax** :FILE[:DIRectory]:DRIVE  
{FLOPy| (SCSI, <NRf> [, <NRf>]) |  
PCCard [, <NRf>] |NETWork|HTTPram|ZIP}  
The first <NRf> of SCSI=SCSI address (0 to 7)  
The second <NRf> of SCSI=Partitions (0 to 2)  
<NRf> of PCCard=Partitions (0 to 1)

**Example** :FILE:DIRectory:DRIVE SCSI, 4, 1

**Description**

- The REALTIME partition (SCSI address=internal ID, partition=0) cannot be specified.
- If the drive does not contain partitions (SCSI or PCCard), omit the <NRf> corresponding to the partitions. "NETWork" can be used when the Ethernet interface (option) is installed.

**:FILE[:DIRectory]:FILTEr**

**Function** Sets a filter to the current file or queries the current setting.

**Syntax** :FILE[:DIRectory]:FILTEr {ITEM|ALL}  
:FILE[:DIRectory]:FILTEr?

**Example** :FILE:DIRectory:FILTEr ITEM  
:FILE:DIRectory:FILTEr? -> :FILE:  
DIRectory:FILTEr ITEM

**:FILE[:DIRectory]:FREE?**

**Function** Queries the free disk space (bytes) on the current storage medium.

**Syntax** :FILE[:DIRectory]:FREE?

**Example** :FILE:DIRectory:FREE? -> :FILE:  
DIRectory:FREE 3.7567939E+09

**:FILE[:DIRectory]:MDIRectory (Make  
Directory)**

**Function** Creates a directory in the current directory. This is an overlap command.

**Syntax** :FILE[:DIRectory]:MDIRectory  
<String>

**Example** :FILE:DIRectory:MDIRectory "NO\_1"

**:FILE[:DIRectory]:PATH?**

**Function** Queries the current directory.

**Syntax** :FILE[:DIRectory]:PATH?

**Example** :FILE:DIRectory:PATH? -> :FILE:  
DIRectory:PATH "Path=HD-0"

**:FILE:LOAD:{BINARy|REALtime|SETup|  
SNAP}:ABORt**

**Function** Aborts the loading of the data.

**Syntax** :FILE:LOAD:{BINARy|REALtime|SETup|  
SNAP}:ABORt

**Example** (The following is an example for the setup data.)  
:FILE:LOAD:SETup:ABORt

**:FILE:LOAD:{BINARy|REALtime|SETup|  
SNAP}[:EXECute]**

**Function** Executes the loading of the data. This is an overlap command.

**Syntax** :FILE:LOAD:{BINARy|REALtime|SETup|  
SNAP}[:EXECute] <Filename>

**Example** (The following is an example for the setup data.)  
:FILE:LOAD:SETup:EXECUTE "CASE1"

**Description**

- The target medium to be loaded is selected using :FILE[:DIRectory]:DRIVE.
- This command cannot be executed on data on the REALTIME partition (SCSI address=internal ID, partition=0).

**:FILE:SAVE?**

**Function** Queries all settings related to the saving of files.

**Syntax** :FILE:SAVE?

**Example** :FILE:SAVE? -> :FILE:SAVE:BINARy:  
TRACE ALL;RANGE MAIN;HISTORY ONE;  
COMPRESSION 0;:FILE:SAVE:ASCIi:  
TRACE ALL;RANGE MAIN;HISTORY ONE;  
INTERVAL OFF;:FILE:SAVE:FLOAT:  
TRACE ALL;RANGE MAIN;HISTORY ONE;:  
FILE:SAVE:ANAMING OFF;NAME "TTL";  
COMMENT ""



## 6.12 FILE Group

### **:FILE:SAVE:ANAMing**

Function Sets the auto naming function of the files to be saved or queries the current setting.

Syntax :FILE:SAVE:ANAMing  
{DATE|NUMBERing|OFF}  
:FILE:SAVE:ANAMing?

Example :FILE:SAVE:ANAMING DATE  
:FILE:SAVE:ANAMING? -> :FILE:SAVE:  
ANAMING DATE

### **:FILE:SAVE:{ASCIi|BINary|FLOat}?**

Function Queries all settings related to the saving of the data.

Syntax :FILE:SAVE:{ASCIi|BINary|FLOat}?

Example :FILE:SAVE:BINARY? -> :FILE:SAVE:  
BINARY:TRACE ALL;RANGE MAIN;  
HISTORY ONE;COMPRESSION 0

### **:FILE:SAVE:{ASCIi|BINary|FLOat|MEASure|SETup|SNAP}:ABORT**

Function Aborts the saving of the data.

Syntax :FILE:SAVE:{ASCIi|BINary|FLOat|  
MEASure|SETup|SNAP}:ABORT

Example (The following is an example for the waveform  
data.)  
:FILE:SAVE:BINARY:ABORT

### **:FILE:SAVE:{ASCIi|BINary|FLOat|MEASure|SETup|SNAP}[:EXECute]**

Function Executes the saving of the data to a file. This is an overlap command.

Syntax :FILE:SAVE:{ASCIi|BINary|FLOat|  
MEASure|SETup|SNAP}[:EXECute]

Example (The following is an example for the waveform  
data.)  
:FILE:SAVE:BINARY:EXECUTE

### **:FILE:SAVE:{ASCIi|BINary|FLOat}**

#### **:HISTory**

Function Sets the number of history pages of the saved data or queries the current setting.

Syntax :FILE:SAVE:{ASCIi|BINary|FLOat}:  
HISTory {ONE|ALL|AVE}  
:FILE:SAVE:{ASCIi|BINary|FLOat}:  
HISTory?

Example :FILE:SAVE:BINARY:HISTORY ALL  
:FILE:SAVE:BINARY:HISTORY? ->  
:FILE:SAVE:BINARY:HISTORY ALL

### **:FILE:SAVE:{ASCIi|BINary|FLOat}**

#### **:RANGe**

Function Sets the range of data to be saved or queries the current setting.

Syntax :FILE:SAVE:{ASCIi|BINary|FLOat}:  
RANGe {MAIN|Z1|Z2}  
:FILE:SAVE:{ASCIi|BINary|FLOat}:  
RANGe?

Example :FILE:SAVE:BINARY:RANGE Z1  
:FILE:SAVE:BINARY:RANGE? -> :FILE:  
SAVE:BINARY:RANGE Z1

### **:FILE:SAVE:{ASCIi|BINary|FLOat}**

#### **:TRACe**

Function Sets the waveform to be saved or queries the current setting.

Syntax :FILE:SAVE:{ASCIi|BINary|FLOat}:  
TRACe{ALL|<Nrf>|MATH[<x>]|PODA|PODB|  
EVENT}  
:FILE:SAVE:{ASCIi|BINary|FLOat}:  
TRACe?  
<Nrf>=1 to 16  
<x>=1 to 8

Example :FILE:SAVE:ASCII:TRACE 1  
:FILE:SAVE:ASCII:TRACE? -> :FILE:  
SAVE:ASCII:TRACE 1

### **:FILE:SAVE:ASCIi:INTERval**

Function Sets the decimation interval for saving data or queries the current setting.

Syntax :FILE:SAVE:ASCIi:INTERval  
{<Nrf>|OFF}  
:FILE:SAVEASCIi:INTERval?  
<Nrf>=5, 10, 20, 50, 100, 200, 500, 1000,  
2000, or 5000

Example :FILE:SAVE:ASCIi:INTERVAL 2000  
:FILE:SAVE:ASCIi:INTERVAL? ->  
:FILE:SAVE:ASCIi:INTERVAL 2000

### **:FILE:SAVE:BINary:COMPRESSion**

Function Sets the data compression of the waveform to be saved or queries the current setting.

Syntax :FILE:SAVE:BINary:COMPRESSion  
{<Boolean>}  
:FILE:SAVE:BINary:COMPRESSion?

Example :FILE:SAVE:BINARY:COMPRESSION ON  
:FILE:SAVE:BINARY:COMPRESSION? ->  
:FILE:SAVE:BINARY:COMPRESSION 1

Description • The range to be saved is Main screen only.  
• Only binary data can be compressed.

### **:FILE:SAVE:COMMENT**

Function Sets the comment of data to be saved or queries the current setting.

Syntax :FILE:SAVE:COMMENT <String>  
:FILE:SAVE:COMMENT?  
<String>=Up to 25 characters

Example :FILE:SAVE:  
COMMENT "THIS IS TEST.        "  
:FILE:SAVE:COMMENT? -> :FILE:SAVE:  
COMMENT "THIS IS TEST.        "

Description Only the characters and symbols displayed on the keyboard on the screen can be used. In ASCII code Ω and Ιsm are expressed as 1EH and 1FH, respectively.

### **:FILE:SAVE:NAME**

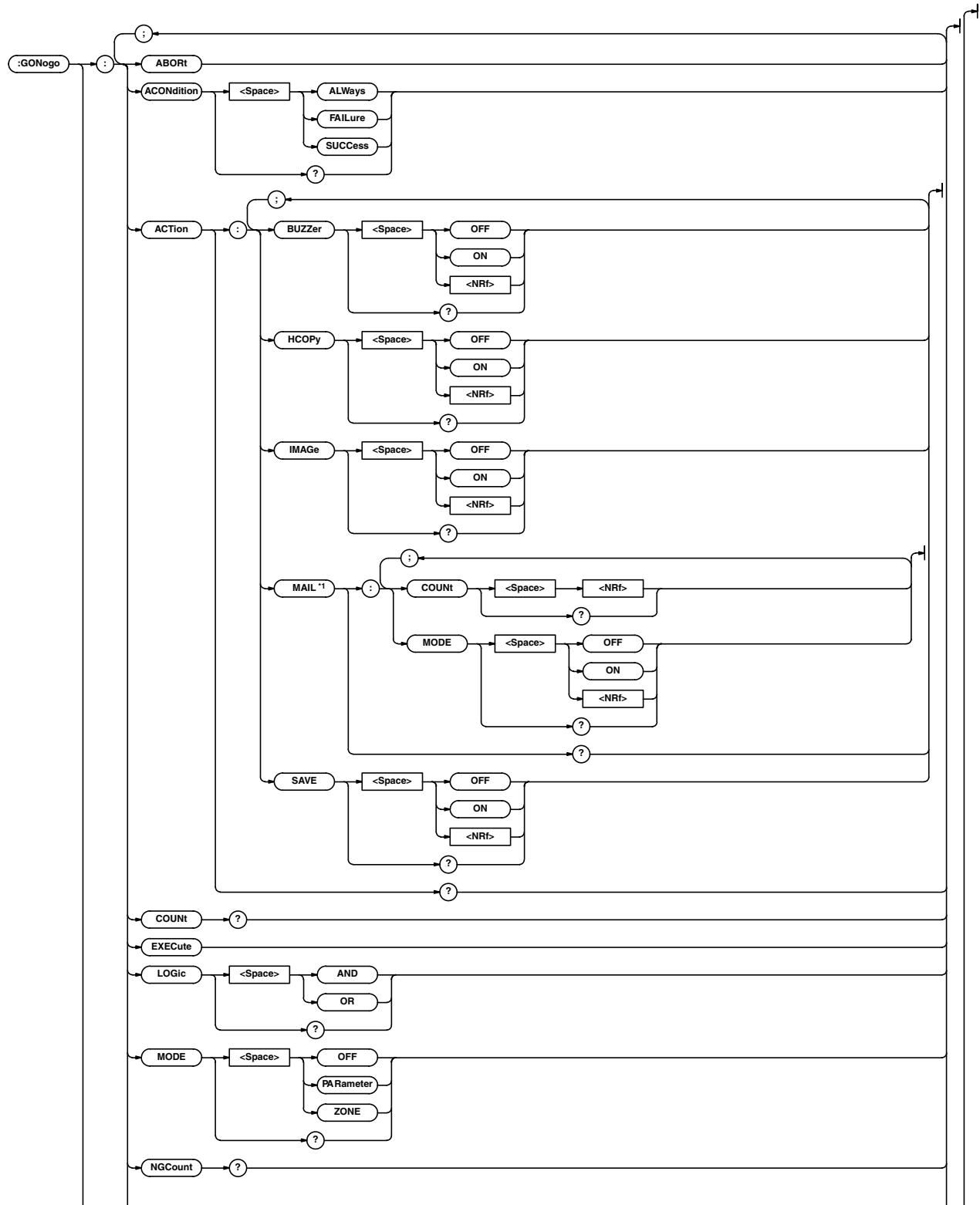
Function Sets the name of the file to be saved or queries the current setting.

Syntax :FILE:SAVE:NAME <Filename>  
:FILE:SAVE:NAME?

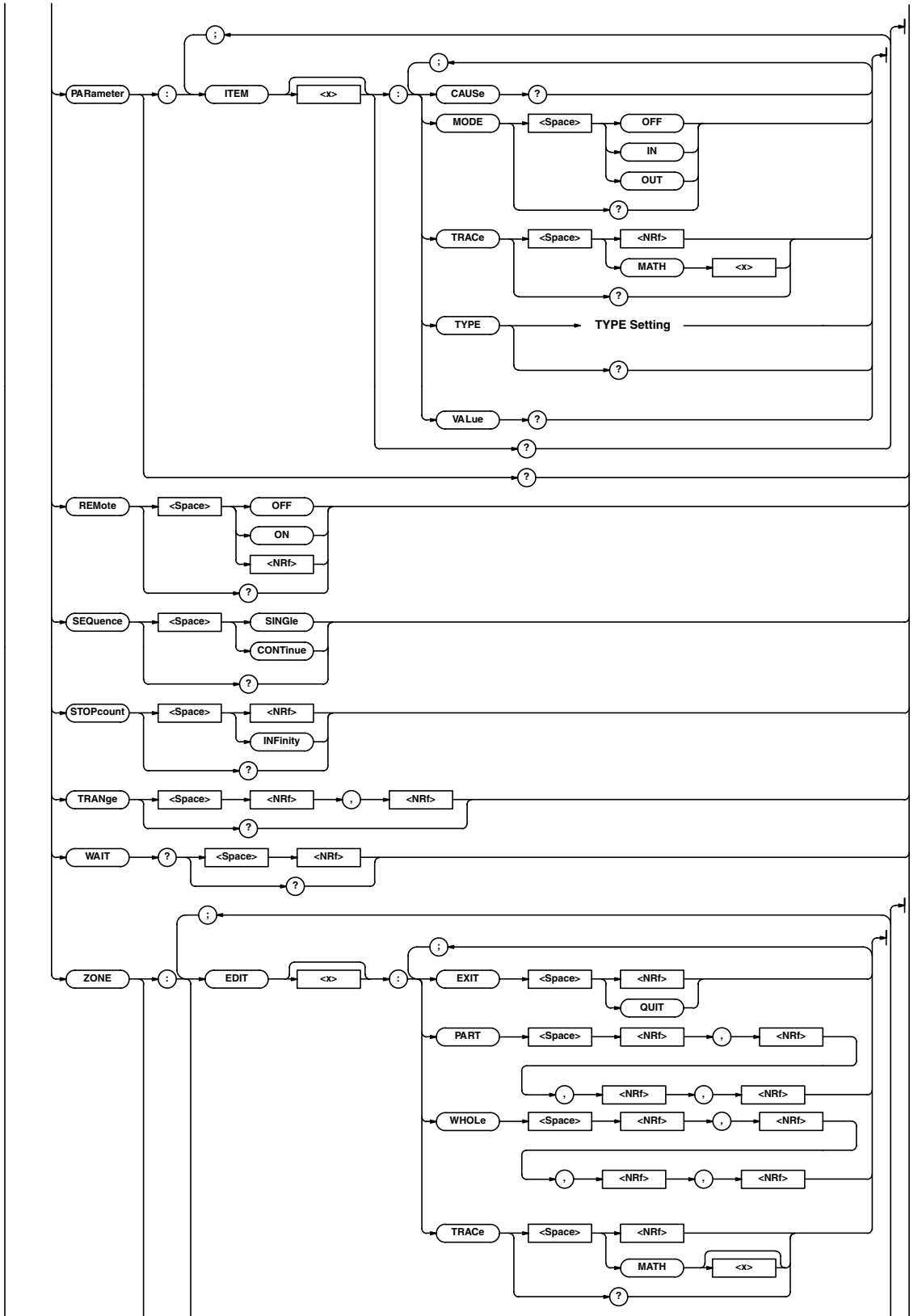
Example :FILE:SAVE:NAME "CASE1"  
:FILE:SAVE:NAME? -> :FILE:SAVE:  
NAME "CASE1"

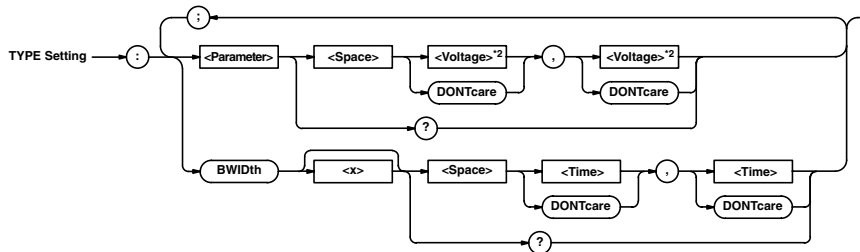
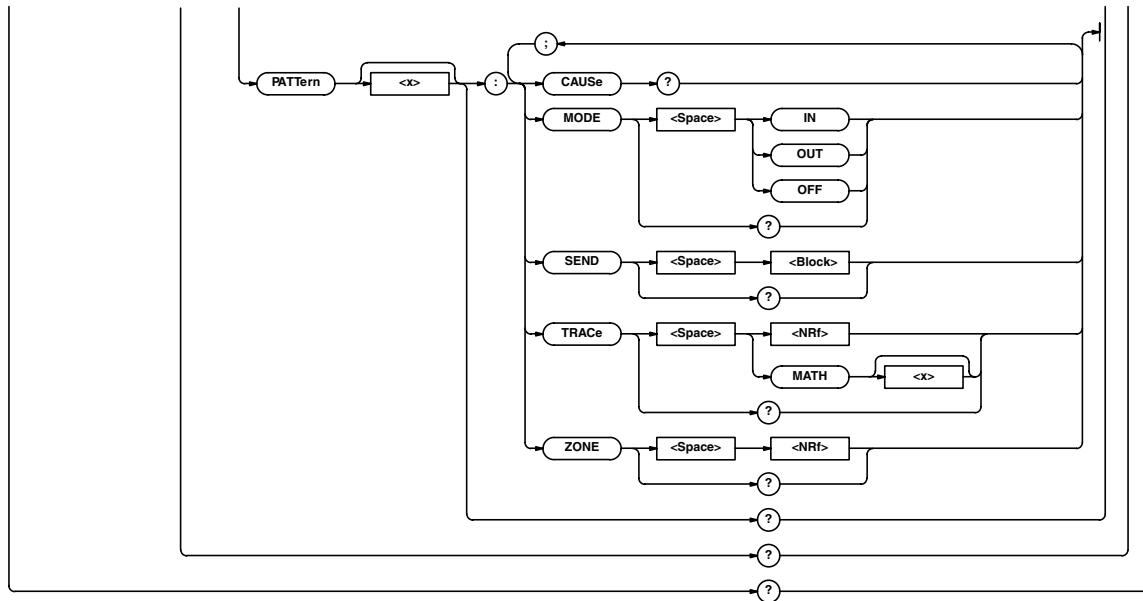
### 6.13 GONogo Group

The commands in this group deal with the GO/NO-GO determination. You can make the same settings and inquiries of settings and measured values as when the (SHIFT+)Measure key on the front panel is used. When the time axis range setting is as follows, the selectable range varies depending on the record length setting. For details, see "Selectable Range of Cursor Position" in section 8.1 of the DL750 User's Manual.  
 <NRf>=-5 to 5 div



6.13 GONogo Group





\*1 Can be used when the Ethernet interface (option) is installed.  
 \*2 <Voltage> when <Parameter> is AMPLitude, AVERAge, HIGH, LOW, MAXimum, MIDDLE, MINimum, PTOPeak, RMS, or SDEVIation (except when using a current probe, then it is <Current>). When the coupling is set to TC or when linear scaling is used, it is <NRF>; <Time> when <Parameter> is AVGPPeriod, BWIDth1, BWIDth2, DELay, FALL, NWIDth, PERiod, PWIDth, or RISE; <Frequency> when <Parameter> is AVGFreq or FREQUency; and <NRF> when <Parameter> is DUTYcycle, NOVershoot, PNUMber, POVershoot, TY1Integ, TY2Integ, XY1Integ, or XY2Integ.

## 6.13 GONogo Group

### :GONogo?

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

**Syntax** :GONogo?

**Example** :GONOGO? -> :GONOGO:MODE PARAMETER;  
PARAMETER:ITEM1:MODE OUT;TRACE 1;  
TYPE:AVERAGE 100.000E-03,  
-100.000E-03;HIGH 100.000E-03,  
-100.000E-03;:GONOGO:PARAMETER:  
ITEM2:MODE OFF;TRACE 1;TYPE:  
MINIMUM DONTCARE,DONTCARE;:GONOGO:  
PARAMETER:ITEM3:MODE OFF;TRACE 1;  
TYPE:RMS DONTCARE,DONTCARE;:GONOGO:  
PARAMETER:ITEM4:MODE OFF;TRACE 1;  
TYPE:AVERAGE DONTCARE,DONTCARE;  
HIGH DONTCARE,DONTCARE;:GONOGO:  
PARAMETER:ITEM5:MODE OFF;TRACE 1;  
TYPE:PTOPEAK DONTCARE,DONTCARE;:  
GONOGO:PARAMETER:ITEM6:MODE OFF;  
TRACE 1;TYPE:PTOPEAK DONTCARE,  
DONTCARE;:GONOGO:PARAMETER:ITEM7:  
MODE OFF;TRACE 1;TYPE:  
PTOPEAK DONTCARE,DONTCARE;:GONOGO:  
PARAMETER:ITEM8:MODE OFF;TRACE 1;  
TYPE:PTOPEAK DONTCARE,DONTCARE;:  
GONOGO:PARAMETER:ITEM9:MODE OFF;  
TRACE 1;TYPE:PTOPEAK DONTCARE,  
DONTCARE;:GONOGO:PARAMETER:ITEM10:  
MODE OFF;TRACE 1;TYPE:  
PTOPEAK DONTCARE,DONTCARE;:GONOGO:  
PARAMETER:ITEM11:MODE OFF;TRACE 1;  
TYPE:PTOPEAK DONTCARE,DONTCARE;:  
GONOGO:PARAMETER:ITEM12:MODE OFF;  
TRACE 1;TYPE:PTOPEAK DONTCARE,  
DONTCARE;:GONOGO:PARAMETER:ITEM13:  
MODE OFF;TRACE 1;TYPE:  
PTOPEAK DONTCARE,DONTCARE;:GONOGO:  
PARAMETER:ITEM14:MODE OFF;TRACE 1;  
TYPE:PTOPEAK DONTCARE,DONTCARE;:  
GONOGO:PARAMETER:ITEM15:MODE OFF;  
TRACE 1;TYPE:PTOPEAK DONTCARE,  
DONTCARE;:GONOGO:PARAMETER:ITEM16:  
MODE OFF;TRACE 1;TYPE:  
PTOPEAK DONTCARE,DONTCARE;:GONOGO:  
ZONE:EDIT1:TRACE1;:GONOGO:ZONE:EDIT2:  
TRACE 1;:GONOGO:ZONE:EDIT3:  
TRACE 1;:GONOGO:ZONE:EDIT4:  
TRACE 1;:GONOGO:ZONE:EDIT5:  
TRACE 1;:GONOGO:ZONE:EDIT6:  
TRACE 1;:GONOGO:ZONE:PATTERN1:  
MODE OFF;TRACE 1;ZONE 0;:GONOGO:  
ZONE:PATTERN2:MODE OFF;TRACE 1;  
ZONE 0;:GONOGO:ZONE:PATTERN3:  
MODE OFF;TRACE 1;ZONE 0;:GONOGO:  
ZONE:PATTERN4:MODE OFF;TRACE 1;  
ZONE 0;:GONOGO:ZONE:PATTERN5:  
MODE OFF;TRACE 1;ZONE 0;:GONOGO:  
ZONE:PATTERN6:MODE OFF;TRACE 1;  
ZONE 0;:GONOGO:ZONE:PATTERN7:  
MODE OFF;TRACE 1;ZONE 0;:GONOGO:  
ZONE:PATTERN8:MODE OFF;TRACE 1;  
ZONE 0;:GONOGO:ZONE:PATTERN9:  
MODE OFF;TRACE 1;ZONE 0;:GONOGO:  
ZONE:PATTERN10:MODE OFF;TRACE 1;  
ZONE 0;:GONOGO:ZONE:PATTERN11:  
MODE OFF;TRACE 1;ZONE 0;:GONOGO:  
ZONE:PATTERN12:MODE OFF;TRACE 1;  
ZONE 0;:GONOGO:ZONE:PATTERN13:

MODE OFF;TRACE 1;ZONE 0;:GONOGO:  
ZONE:PATTERN14:MODE OFF;TRACE 1;  
ZONE 0;:GONOGO:ZONE:PATTERN15:  
MODE OFF;TRACE 1;ZONE 0;:GONOGO:  
ZONE:PATTERN16:MODE OFF;TRACE 1;  
ZONE 0;:GONOGO:ACTION:HCOPIY 0;  
IMAGE 0;SAVE 0;BUZZER 0;MAIL:  
MODE 0;COUNT 100;:GONOGO:  
TRANGE -5.0000000,5.0000000;  
LOGIC AND;ACONDITION SUCCESS;  
SEQUENCE SINGLE;REMOTE 0;  
STOPCOUNT INFINITY

### :GONogo:ABORt

**Function** Aborts the GO/NO-GO determination.

**Syntax** :GONogo:ABORt

**Example** :GONOGO:ABORt

### :GONogo:ACONdition

**Function** Sets the criteria for the action of the GO/NO-GO determination or queries the current setting.

**Syntax** :GONogo:ACONdition  
{ALWayS|FAILure|SUCCeSS}  
:GONogo:ACONdition?

**Example** :GONOGO:ACONDITION FAILURE  
:GONOGO:ACONDITION? -> :GONOGO:  
ACONDITION FAILURE

### :GONogo:ACTion?

**Function** Queries all settings related to the action taken when the execution condition is met.

**Syntax** :GONogo:ACTion?

**Example** :GONOGO:ACTION? -> :GONOGO:ACTION:  
HCOPIY 0;IMAGE 0;SAVE 0;BUZZER 0;  
MAIL:MODE 0;COUNT 100

### :GONogo:ACTion:BUZZer

**Function** Sets whether an alarm is sounded when the execution condition is met or queries the current setting.

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

**Example** :GONOGO:ACTION:BUZZER OFF  
:GONOGO:ACTION:BUZZER? -> :GONOGO:  
ACTION:BUZZER 0

### :GONogo:ACTion:HCOPIY (Hard COPY)

**Function** Sets whether to output a hard copy to the built-in printer or an external printer when the execution condition is met or queries the current setting.

**Syntax** :GONogo:ACTion:HCOPIY {<Boolean>}  
:GONogo:ACTion:HCOPIY?

**Example** :GONOGO:ACTION:HCOPIY OFF  
:GONOGO:ACTION:HCOPIY? -> :GONOGO:  
ACTION:HCOPIY 0

### :GONogo:ACTion:IMAGE

**Function** Sets whether to save the screen image to the external storage medium when the execution condition is met or queries the current setting.

**Syntax** :GONogo:ACTion:IMAGE {<Boolean>}  
:GONogo:ACTion:IMAGE?

**Example** :GONOGO:ACTION:IMAGE OFF  
:GONOGO:ACTION:IMAGE? -> :GONOGO:  
ACTION:IMAGE 0

**:GONogo:ACTion:MAIL?**

**Function** Queries all settings related to the mail transmission when the execution condition is met.

**Syntax** :GONogo:ACTion:MAIL?

**Example** :GONOGO:ACTION:MAIL? -> :GONOGO:ACTION:MAIL:MODE 0;COUNT 100

**Description** Can be used when the Ethernet interface (option) is installed.

**:GONogo:ACTion:MAIL:COUNT**

**Function** Sets the upper limit on the number of transmitted mail when the execution condition is met or queries the current setting.

**Syntax** :GONogo:ACTion:MAIL:COUNT {<Nrf>}  
:GONogo:ACTion:MAIL:COUNT?

**Example** :GONOGO:ACTION:MAIL:COUNT 100  
:GONOGO:ACTION:MAIL:COUNT? ->  
:GONOGO:ACTION:MAIL:COUNT 100

**Description** Can be used when the Ethernet interface (option) is installed.

**:GONogo:ACTion:MAIL:MODE**

**Function** Sets whether to send a mail when the execution condition is met or queries the current setting.

**Syntax** :GONogo:ACTion:MAIL:MODE {<Boolean>}  
:GONogo:ACTion:MAIL:MODE?

**Example** :GONOGO:ACTION:MAIL:MODE OFF  
:GONOGO:ACTION:MAIL:MODE? ->  
:GONOGO:ACTION:MAIL:MODE 0

**Description** Can be used when the Ethernet interface (option) is installed.

**:GONogo:ACTion:SAVE**

**Function** Sets whether to save the data to the external storage medium when the execution condition is met or queries the current setting.

**Syntax** :GONogo:ACTion:SAVE {<Boolean>}  
:GONogo:ACTion:SAVE?

**Example** :GONOGO:ACTION:SAVE OFF  
:GONOGO:ACTION:SAVE? -> :GONOGO:ACTION:SAVE 0

**Description** To set or query the storage medium type, use :FILE:DIRECTORY:DRIVE.

**:GONogo:COUNT?**

**Function** Queries the actual number of times GO/NO-GO determination was performed.

**Syntax** :GONogo:COUNT?

**Example** :GONOGO:COUNT? -> :GONOGO:COUNT 10

**:GONogo:EXECute**

**Function** Executes the GO/NO-GO determination. This is an overlap command.

**Syntax** :GONogo:EXECute

**Example** :GONOGO:EXECUTE

**:GONogo:LOGic**

**Function** Sets the criteria logic of the GO/NO-GO determination or queries the current setting.

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

**Example** :GONOGO:LOGIC AND  
:GONOGO:LOGIC? -> :GONOGO:LOGIC AND

**:GONogo:MODE**

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

**Syntax** :GONogo:MODE {OFF|PARAMeter|ZONE}  
:GONogo:MODE?

**Example** :GONOGO:MODE PARAMETER  
:GONOGO:MODE? -> :GONOGO:MODE PARAMETER

**:GONogo:NGCount?**

**Function** Queries the number of NO-GOs of the GO/NO-GO determination.

**Syntax** :GONogo:NGCount?

**Example** :GONOGO:NGCOUNT? -> :GONOGO:NGCOUNT 10

**:GONogo:PARAMeter?**

**Function** Queries all settings related to parameter determination.

**Syntax** :GONogo:PARAMeter?

**Example** :GONOGO:PARAMETER? -> :GONOGO:PARAMETER:ITEM1:MODE OUT;TRACE 1;TYPE:AVERAGE 100.000E-03,-100.000E-03;HIGH 100.000E-03,-100.000E-03;:GONOGO:PARAMETER:ITEM2:MODE OFF;TRACE 1;TYPE:MINIMUM DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM3:MODE OFF;TRACE 1;TYPE:RMS DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM4:MODE OFF;TRACE 1;TYPE:AVERAGE DONTCARE,DONTCARE;HIGH DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM5:MODE OFF;TRACE 1;TYPE:PTOPEAK DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM6:MODE OFF;TRACE 1;TYPE:PTOPEAK DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM7:MODE OFF;TRACE 1;TYPE:PTOPEAK DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM8:MODE OFF;TRACE 1;TYPE:PTOPEAK DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM9:MODE OFF;TRACE 1;TYPE:PTOPEAK DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM10:MODE OFF;TRACE 1;TYPE:PTOPEAK DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM11:MODE OFF;TRACE 1;TYPE:PTOPEAK DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM12:MODE OFF;TRACE 1;TYPE:PTOPEAK DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM13:MODE OFF;TRACE 1;TYPE:PTOPEAK DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM14:MODE OFF;TRACE 1;TYPE:PTOPEAK DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM15:MODE OFF;TRACE 1;TYPE:PTOPEAK DONTCARE,DONTCARE;:GONOGO:PARAMETER:ITEM16:MODE OFF;TRACE 1;TYPE:PTOPEAK DONTCARE,DONTCARE

## 6.13 GONogo Group

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

**Function** Queries all settings related to waveform parameter of the parameter determination.

**Syntax** :GONogo:PARAMeter:ITEM<x>?  
<x> of ITEM<x>=1 to 16

**Example** :GONOGO:PARAMETER:ITEM1? ->  
:GONOGO:PARAMETER:ITEM1:MODE OUT;  
TRACE 1;TYPE:AVERAGE 100.000E-03,  
-100.000E-03;HIGH 100.000E-03,  
-100.000E-03

### :GONogo:PARAMeter:ITEM<x>:CAUSE?

**Function** Queries whether the waveform parameter of the parameter determination is the cause of NO-GO.

**Syntax** :GONogo:PARAMeter:ITEM<x>:CAUSE?  
<x> of ITEM<x>=1 to 16

**Example** :GONOGO:PARAMETER:ITEM1:CAUSE? ->  
:GONOGO:PARAMETER:ITEM1:CAUSE 1

**Description** The query returns a "1" if the parameter is the cause of the NO-GO result or a "0" if it is not.

### :GONogo:PARAMeter:ITEM<x>:MODE

**Function** Turns OFF the waveform parameter of the parameter determination or sets the determination criteria of the waveform parameter or queries the current setting.

**Syntax** :GONogo:PARAMeter:ITEM<x>:MODE  
{OFF|IN|OUT}  
:GONogo:PARAMeter:ITEM<x>:MODE?  
<x> of ITEM<x>=1 to 16

**Example** :GONOGO:PARAMETER:ITEM1:MODE IN  
:GONOGO:PARAMETER:ITEM1:MODE? ->  
:GONOGO:PARAMETER:ITEM1:MODE IN

### :GONogo:PARAMeter:ITEM<x>:TRACE

**Function** Sets the target waveform of the waveform parameter of the parameter determination or queries the current setting.

**Syntax** :GONogo:PARAMeter:ITEM<x>:TRACE  
{<NRF>|MATH<x>}  
:GONogo:PARAMeter:ITEM<x>:TRACE?  
<x> of ITEM<x>=1 to 16  
<NRF>=1 to 16  
<x> of MATH<x>=1 to 8

**Example** :GONOGO:PARAMETER:ITEM1:TRACE 1  
:GONOGO:PARAMETER:ITEM1:TRACE? ->  
:GONOGO:PARAMETER:ITEM1:TRACE 1

### :GONogo:PARAMeter:ITEM<x>:TYPE?

**Function** Queries the item and the upper and lower limits of the waveform parameter of the parameter determination.

**Syntax** :GONogo:PARAMeter:ITEM<x>:TYPE?  
<x> of ITEM<x>=1 to 16

**Example** :GONOGO:PARAMETER:ITEM1:TYPE? ->  
:GONOGO:PARAMETER:ITEM1:TYPE:  
MAXIMUM 1.10000E+00,1.00000E+00

### :GONogo:PARAMeter:ITEM<x>:TYPE:

**<Parameter>**

**Function** Sets the upper and lower limits of the measured item or queries the current setting.

**Syntax** :GONogo:PARAMeter:ITEM<x>:TYPE:  
<Parameter>  
{<{Voltage|DONTcare}>,<{Voltage|DONTcare}>|  
<{Current|DONTcare}>,<{Current|DONTcare}>|  
<{Time|DONTcare}>,<{Time|DONTcare}>|  
<{Frequency|DONTcare}>,<{Frequency|DONTcare}>|  
<{<NRF>|DONTcare}>,<{<NRF>|DONTcare}>}  
:GONogo:PARAMeter:ITEM<x>:TYPE:  
<Parameter>?  
<x> of ITEM<x>=1 to 16  
<Parameter>={AMPLitude|AVERage|  
AVGFreq|AVGPeriod|BWIDth1|BWIDth2|  
DElay|DUTYcycle|FALL|FREquency|HIGH|  
LOW|MAXimum|MIDDLE|MINimum|  
NOVershoot|NWIDTH|PERiod|PNUMBER|  
POVershoot|PTOPeak|PWIDTH|RISE|RMS|  
SDEVIation|TY1Integ|TY2Integ|XY1Integ|  
XY2Integ}  
<Voltage>,<Current>,<Time>,<Frequency>,  
<NRF>=See the *DL750 User's Manual*

**Example** :GONOGO:PARAMETER:ITEM1:TYPE:  
AVERAGE 100MV,-100MV  
:GONOGO:PARAMETER:ITEM1:TYPE:  
AVERAGE? -> :GONOGO:PARAMETER:  
ITEM1:TYPE:AVERAGE 100.000E-03,  
-100.000E-03

**Description** Queries are only allowed on the parameters that have been set using this command.

### :GONogo:PARAMeter:ITEM<x>:VALUE?

**Function** Queries the measured value of the waveform parameter of the parameter determination.

**Syntax** :GONogo:PARAMeter:ITEM<x>:VALUE?  
<x> of ITEM<x>=1 to 16

**Example** :GONOGO:PARAMETER:ITEM1:VALUE? ->  
:GONOGO:PARAMETER:ITEM1:  
VALUE 50.000000E+00

**Description** If the measurement is not possible such as when the mode is set to OFF, "NAN" (Not A Number) is returned.

### :GONogo:REMOte

**Function** Sets the remote mode of the GO/NO-GO determination or queries the current setting.

**Syntax** :GONogo:REMOte {<Boolean>}  
:GONogo:REMOte?

**Example** :GONOGO:REMOTE ON  
:GONOGO:REMOTE? -> :GONOGO:REMOTE 1

### :GONogo:SEQuence

**Function** Sets the action count of the GO/NO-GO determination or queries the current setting.

**Syntax** :GONogo:SEQuence {CONTINUE|SINGLE}  
:GONogo:SEQuence?

**Example** :GONOGO:SEQUENCE CONTINUE  
:GONOGO:SEQUENCE? -> :GONOGO:  
SEQUENCE CONTINUE

**:GONogo:STOPcount**

Function	Sets the waveform acquisition count of the GO/NO-GO determination or queries the current setting.
Syntax	:GONogo:STOPcount {<Nrf> INFinity} :GONogo:STOPcount? <Nrf>=1 to 65536
Example	:GONOGO:STOPCOUNT 10 :GONOGO:STOPCOUNT? -> :GONOGO: STOPCOUNT 10

**:GONogo:TRANGE**

Function	Sets the determination range of GO/NO-GO determination or queries the current setting.
Syntax	:GONogo:TRANGE {<Nrf>, <Nrf>} :GONogo:TRANGE? <Nrf>=-5 to 5 divisions (10 divisions/displayed record length steps)
Example	:GONOGO:TRANGE -5.00,5.00 :GONOGO:TRANGE? -> :GONOGO: TRANGE -5.0000000,5.0000000

**:GONogo:WAIT?**

Function	Waits for the completion of the GO/NO-GO determination.
Syntax	:GONogo:WAIT? {<Nrf>} <Nrf>= 0 to 8640000 (in units of 100 ms)
Example	(Set the timeout to 5 s) :GONOGO:WAIT? 50 -> 0
Description	If a time out occurs, 1 is returned. If the determination completes before the time out, 0 is returned.

**:GONogo:ZONE?**

Function	Queries all settings related to the zone determination.
Syntax	:GONogo:ZONE?
Example	:GONOGO:ZONE? -> :GONOGO:ZONE: EDIT1:TRACE 1;:GONOGO:ZONE:EDIT2: TRACE 1;:GONOGO:ZONE:EDIT3: TRACE 1;:GONOGO:ZONE:EDIT4: TRACE 1;:GONOGO:ZONE:EDIT5: TRACE 1;:GONOGO:ZONE:EDIT6: TRACE 1;:GONOGO:ZONE:PATTERN1: MODE OFF;TRACE 1;ZONE0;:GONOGO: ZONE:PATTERN2:MODE OFF;TRACE 1; ZONE 0;:GONOGO:ZONE:PATTERN3: MODE OFF;TRACE 1;ZONE 0;:GONOGO: ZONE:PATTERN4:MODE OFF;TRACE 1; ZONE 0;:GONOGO:ZONE:PATTERN5: MODE OFF;TRACE 1;ZONE 0;:GONOGO: ZONE:PATTERN6:MODE OFF;TRACE 1; ZONE 0;:GONOGO:ZONE:PATTERN7: MODE OFF;TRACE 1;ZONE 0;:GONOGO: ZONE:PATTERN8:MODE OFF;TRACE 1; ZONE 0;:GONOGO:ZONE:PATTERN9: MODE OFF;TRACE 1;ZONE 0;:GONOGO: ZONE:PATTERN10:MODE OFF;TRACE 1; ZONE 0;:GONOGO:ZONE:PATTERN11: MODE OFF;TRACE 1;ZONE 0;:GONOGO: ZONE:PATTERN12:MODE OFF;TRACE 1; ZONE 0;:GONOGO:ZONE:PATTERN13: MODE OFF;TRACE 1;ZONE 0;:GONOGO: ZONE:PATTERN14:MODE OFF;TRACE 1; ZONE 0;:GONOGO:ZONE:PATTERN15: MODE OFF;TRACE 1;ZONE 0;:GONOGO: ZONE:PATTERN16:MODE OFF;TRACE 1; ZONE 0

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

Function	Exits from the edit menu of the zone for zone determination.
Syntax	:GONogo:ZONE:EDIT<x>:EXIT {<Nrf> QUIT} <x> of Edit <x>=1 to 12 (1 to 6 are edit zones 1 to 6 for a new zone waveform and 7 to 12 are edit zones 1 to 6 for an existing zone waveform) <Nrf>=1 to 6 (destination zone No. for saving the zone being edited)
Example	:GONOGO:ZONE:EDIT1:EXIT 1
Description	<ul style="list-style-type: none"> <li>If you set the value to "QUIT," the zone specified by :GONogo:ZONE:EDIT&lt;x&gt;:{PART WHOLE} is cancelled. If you wish to save the zone specified by :GONogo:ZONE:EDIT&lt;x&gt;:{PART WHOLE}, be sure to set the value of this command to 1 to 6.</li> <li>If you do not execute this command, the zone specified by :GONogo:ZONE:EDIT&lt;x&gt;:{PART WHOLE} will be cancelled.</li> <li>An error occurs if this command is issued when the zone is not being edited.</li> </ul>

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

Function	Edits a portion of the zone for zone determination.
Syntax	:GONogo:ZONE:EDIT<x>:PART {<Nrf>, <Nrf>, <Nrf>, <Nrf>} <x> of Edit <x>=1 to 12 (1 to 6 are edit zones 1 to 6 for a new zone waveform and 7 to 12 are edit zones 1 to 6 for an existing zone waveform) <Nrf>=-5.00 to 5.00 (div, 0.01 div steps: T1 and T2 cursors) -10.00 to 10.00 (div, 0.01 steps: up and down) (In the order {T1, T2, ↑, and ↓})
Example	:GONOGO:ZONE:EDIT1: PART -2.50,2.50,1.00,1.00
Description	<ul style="list-style-type: none"> <li>When you are done editing, exit from the edit mode using :GONogo:ZONE:EDIT&lt;x&gt;:EXIT {&lt;NRF&gt;}.</li> <li>If you do not execute :GONogo:ZONE:EDIT&lt;x&gt;:EXIT {&lt;NRF&gt;}, the edited zone will not be saved.</li> </ul>



## 6.13 GONogo Group

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

**Function** Edits the zone for zone determination.

**Syntax** :GONogo:ZONE:EDIT<x>:WHOLE  
{<NRf>,<NRf>,<NRf>,<NRf>}  
<x> of Edit <x>=1 to 12 (1 to 6 are edit zones 1 to 6 for a new zone waveform and 7 to 12 are edit zones 1 to 6 for an existing zone waveform)  
<NRf>=0 to 5.00 (div, 0.01 steps: left and right)  
0 to 10.00 (div, 0.01 steps: up and down)  
(In the order {←, →, ↑, and ↓})

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

**Description**

- When you are done editing, exit from the edit mode using  
:GONogo:ZONE:EDIT<x>:EXIT {<NRf>}.
- If you do not execute  
:GONogo:ZONE:EDIT<x>:EXIT {<NRf>}, the edited zone will not be saved.

### :GONogo:ZONE:EDIT<x>:TRACe

**Function** Sets the target channel for the new zone waveform or queries the current setting.

**Syntax** :GONogo:ZONE:EDIT<x>:TRACe  
{<NRf>|MATH<y>}  
:GONogo:ZONE:EDIT<x>:TRACe?  
<x> of EDIT<x>=1 to 6  
<NRf>=1 to 16  
<y> of MATH<y>=1 to 8

**Example** :GONOGO:ZONE:EDIT1:TRACe 1  
:GONOGO:ZONE:EDIT1:TRACe? ->  
:GONOGO:ZONE:EDIT1:TRACe 1

**Description** The target waveform is valid only for new zone waveforms. Thus, 7 to 12 are not available.

### :GONogo:ZONE:PATtern<x>?

**Function** Queries all settings related to the zone determination of the specified pattern.

**Syntax** :GONogo:ZONE:PATtern<x>?  
<x>=1 to 16

**Example** :GONOGO:ZONE:PATTERN1? -> :GONOGO:  
ZONE:PATTERN1:MODE OFF;TRACE 1;  
ZONE 0

### :GONogo:ZONE:PATtern<x>:CAUSE?

**Function** Queries whether the pattern of the zone determination is the cause of NO-GO.

**Syntax** :GONogo:ZONE:PATtern<x>:CAUSE?  
<x>=1 to 16

**Example** :GONOGO:ZONE:PATTERN1:CAUSE? ->  
:GONOGO:ZONE:PATTERN1:CAUSE 1

**Description** The query returns a "1" if the determination zone is the cause of the NO-GO result or a "0" if it is not.

### :GONogo:ZONE:PATtern<x>:MODE

**Function** Turns OFF the pattern of zone determination or sets the determination criteria or queries the current setting.

**Syntax** :GONogo:ZONE:PATtern<x>:MODE  
{IN|OUT|OFF}  
:GONogo:ZONE:PATtern<x>:MODE?  
<x>=1 to 16

**Example** :GONOGO:ZONE:PATTERN1:MODE IN  
:GONOGO:ZONE:PATTERN1:MODE? ->  
:GONOGO:ZONE:PATTERN1:MODE IN

### :GONogo:ZONE:PATtern<x>:SEND

**Function** Sets the zone data specified for the determination zone or queries the current setting.

**Syntax** :GONogo:ZONE:PATtern<x>:SEND  
#6004028  
<data byte sequence>  
:GONnogo:ZONE:PATtern<x>:SEND?  
<x>=1 to 16

**Example** :GONOGO:ZONE:PATTERN1:SEND #6004028  
<data byte sequence>  
:GONOGO:ZONE:PATTERN1:SEND? ->  
:GONOGO:ZONE:PATTERN1:SEND #6004028  
<data byte sequence>

**Description**

- The (data byte sequence) consists of zone data of length 4028 bytes.
- Zone data consists of 8-bit data (0 to 255). This data string may contain ASCII codes "0AH" or "00H" that stand for "NL." Hence, care must be taken when programming the controller.
- This command is used to save the zone data to the PC and set the data as zone data on the instrument.
- When this command is used to set the zone data to the instrument for performing the determination, issue  
:GONogo:ZONE:PATtern<x>:MODE afterwards.

### :GONogo:ZONE:PATtern<x>:TRACe

**Function** Sets the target waveform for the pattern of the zone determination or queries the current setting.

**Syntax** :GONogo:ZONE:PATtern<x>:TRACe  
{<NRf>|MATH<x>}  
:GONogo:ZONE:PATtern<x>:TRACe?  
<NRf>=1 to 16  
<x> of <PATtern x>=1 to 16  
<x> of MATH<x>=1 to 8

**Example** :GONOGO:ZONE:PATTERN1:TRACe 1  
:GONOGO:ZONE:PATTERN1:TRACe? ->  
:GONOGO:ZONE:PATTERN1:TRACe 1

### :GONogo:ZONE:PATtern<x>:ZONE

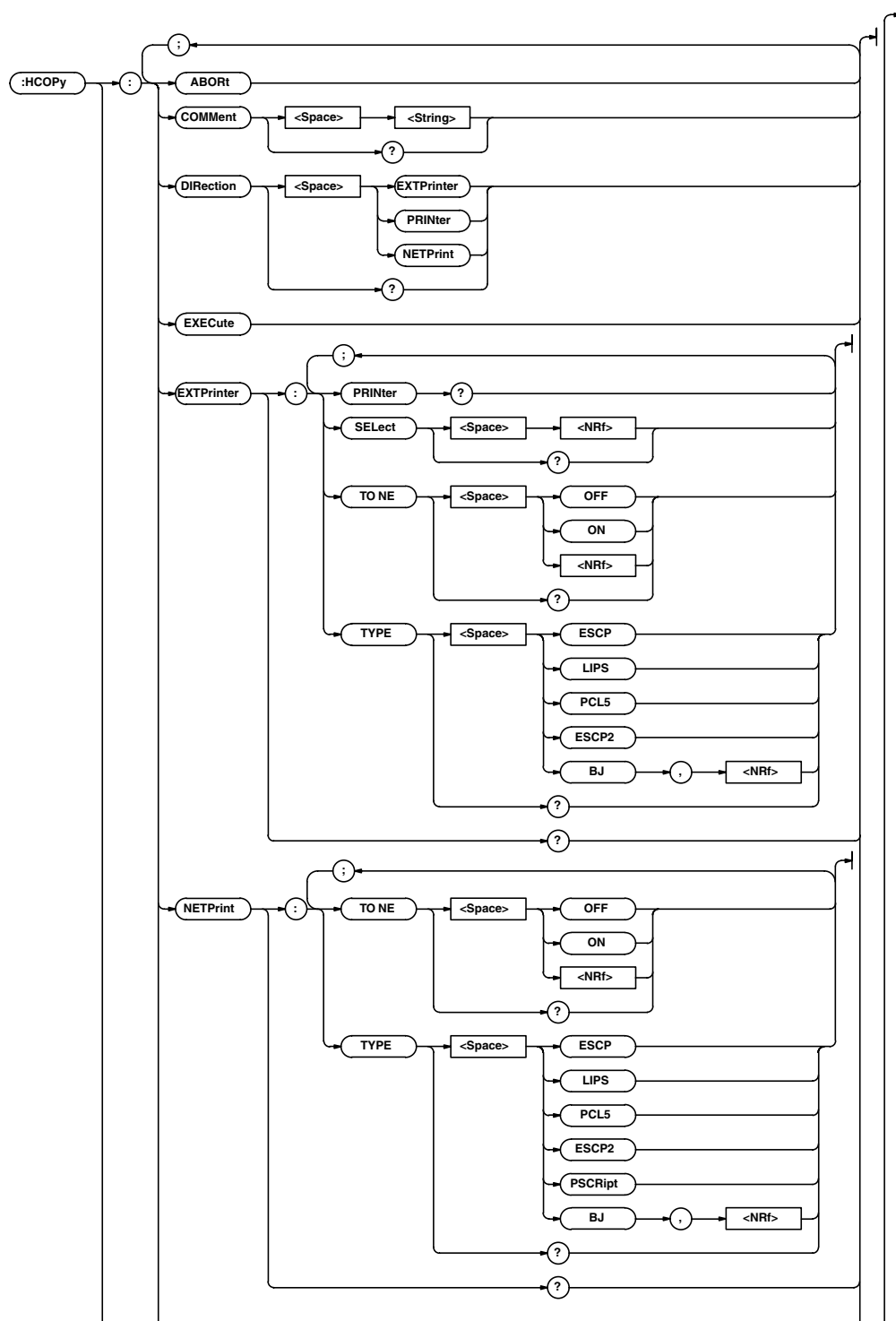
**Function** Sets the target zone data of the pattern of the zone determination or queries the current setting.

**Syntax** :GONogo:ZONE:PATtern<x>:ZONE  
{<NRf>}  
:GONogo:ZONE:PATtern<x>:ZONE?  
<NRf>=0 to 5

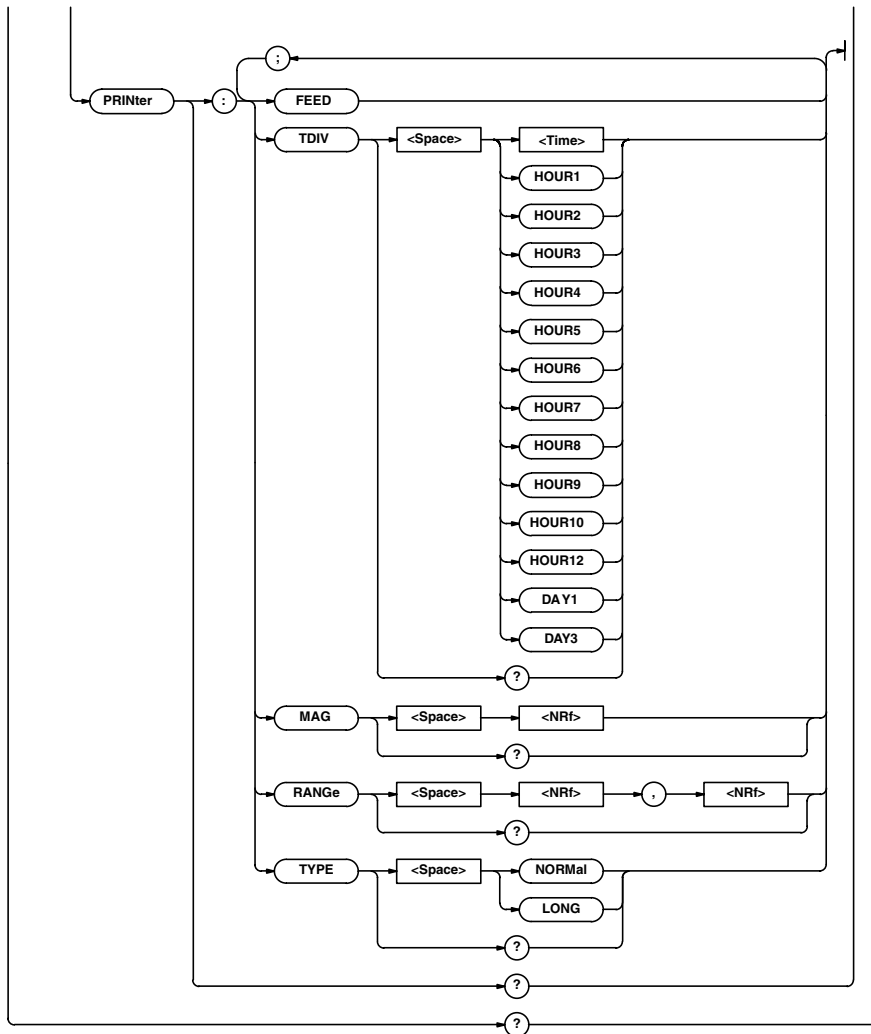
**Example** :GONOGO:ZONE:PATTERN1:ZONE 0  
:GONOGO:ZONE:PATTERN1:ZONE? ->  
:GONOGO:ZONE:PATTERN1:ZONE 0

## 6.14 HCOPY Group

The commands in this group deal with the output of screen data to the built-in printer and other printers. You can make the same settings, execute the same operations, and make the same inquiries as when the (SHIFT+)PRINT key on the front panel is used.



## 6.14 HCOpy Group



### **:HCOpy? (Hard COPY)**

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

**Syntax** :HCOpy?

**Example** :HCOpy? -> :HCOpy:  
 DIRECTION PRINTER;PRINter:  
 TYPE LONG;MAG 20.00E-03;  
 RANGE -5.0000000,5.0000000;:HCOpy:  
 COMMENT "HCOpy COMMENT";EXTPRINter:  
 TYPE PCL5;TONE 1;SELECT NAN;:HCOpy:  
 NETPRINT:TYPE ESCP2;TONE 1

### **:HCOpy:ABORt**

**Function** Aborts data output and paper feeding.

**Syntax** :HCOpy:ABORt

**Example** :HCOpy:ABORt

### **:HCOpy:COMMeNt**

**Function** Sets the comment on the lower right corner of the screen or queries the current setting.

**Syntax** :HCOpy:COMMeNt <String>  
 :HCOpy:COMMeNt?  
 <String>=Up to 20 characters

**Example** :HCOpy:COMMeNt "THIS IS TEST."  
 :HCOpy:COMMeNt? -> :HCOpy:  
 COMMENT "THIS IS TEST."

### **:HCOpy:DIReCtion**

**Function** Sets the data output destination (print destination) or queries the current setting.

**Syntax** :HCOpy:DIReCtion  
 {EXTPRINter|PRINter|NETPrint}  
 :HCOpy:DIReCtion?

**Example** :HCOpy:DIReCtion PRINter  
 :HCOpy:DIReCtion? -> :HCOpy:  
 DIRECTION PRINter

**Description** "NETPrint" can be used when the Ethernet interface (optional) is installed.

### **:HCOpy:EXECCute**

**Function** Executes the data output. This is an overlap command.

**Syntax** :HCOpy:EXECCute

**Example** :HCOpy:EXECCute

### **:HCOpy:EXTPRINter?**

**Function** Queries all settings related to the external printer output.

**Syntax** :HCOpy:EXTPRINter?

**Example** :HCOpy:EXTPRINter? -> :HCOpy:  
 EXTPRINter:TYPE PCL5;TONE 1;  
 SELECT NAN

**:HCOPY:EXTPrinter:PRINTER?**

**Function** Outputs the addresses of all the printers that are currently connected.

**Syntax** :HCOPY:EXTPrinter:PRINTER?

**Example** (The following is an example when printers are connected to address 2 and 3.)  
:HCOPY:EXTPRINTER:PRINTER? ->  
:HCOPY:EXTPRINTER:PRINTER 2,3

**:HCOPY:EXTPrinter:SElect**

**Function** Sets the address of the output printer or queries the current setting.

**Syntax** :HCOPY:EXTPrinter:SElect {<NRf>}  
:HCOPY:EXTPrinter:SElect?

**Example** :HCOPY:EXTPRINTER:SELECT 3  
:HCOPY:EXTPRINTER:SELECT? ->  
:HCOPY:EXTPRINTER:SELECT 3

**Description** If you specify an address where no printer is connected, an error occurs. If a query is made to an address where no printer is connected, "NAN (Not A Number)" is returned.

**:HCOPY:EXTPrinter:TONE**

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

**Syntax** :HCOPY:EXTPrinter:TONE {<Boolean>}  
:HCOPY:EXTPrinter:TONE?

**Example** :HCOPY:EXTPRINTER:TONE ON  
:HCOPY:EXTPRINTER:TONE? -> :HCOPY:  
EXTPRINTER:TONE 1

**:HCOPY:EXTPrinter:TYPE**

**Function** Sets the type of output commands to send to the external printer or queries the current setting.

**Syntax** :HCOPY:EXTPrinter:TYPE  
{BJ,<NRf>|ESCP|LIPS|PCL5|ESCP2}  
:HCOPY:EXTPrinter:TYPE?  
<NRf>=180, 300, or 360 (dpi)

**Example** :HCOPY:EXTPRINTER:TYPE BJ,360  
:HCOPY:EXTPRINTER:TYPE? -> :HCOPY:  
EXTPRINTER:TYPE BJ,360

**Description** <NRf> is specified only when the type is set to "BJ."

**:HCOPY:NETPrint?**

**Function** Queries all settings related to the network printer output.

**Syntax** :HCOPY:NETPrint?

**Example** :HCOPY:NETPRINT? -> :HCOPY:  
NETPRINT:TYPE POSTSCRIPT;TONE 1

**Description** Can be used when the Ethernet interface (option) is installed.

**:HCOPY:NETPrint:TONE**

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

**Syntax** :HCOPY:NETPrint:TONE {<Boolean>}  
:HCOPY:NETPrint:TONE?

**Example** :HCOPY:NETPRINT:TONE ON  
:HCOPY:NETPRINT:TONE? -> :HCOPY:  
NETPRINT:TONE 1

**Description** Can be used when the Ethernet interface (option) is installed.

**:HCOPY:NETPrint:TYPE**

**Function** Sets the type of output commands to send to the network printer or queries the current setting.

**Syntax** :HCOPY:NETPrint:TONE {BJ,  
<NRf>|ESCP|LIPS|PCL5|ESCP2|PSCRIPT}  
:HCOPY:NETPrint:TYPE?  
<NRf>=180, 300, or 360 (dpi)

**Example** :HCOPY:NETPRINT:TYPE PSCRIPT  
:HCOPY:NETPRINT:TYPE? -> :HCOPY:  
NETPRINT:TYPE PSCRIPT

**Description** • <NRf> is specified only when the type is set to "BJ."  
• Can be used when the Ethernet interface (option) is installed.

**:HCOPY:PRINTER?**

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

**Syntax** :HCOPY:PRINTER?

**Example** :HCOPY:PRINTER? -> :HCOPY:PRINTER:  
TYPE LONG;MAG 20.00E-03;  
RANGE -5.0000000,5.0000000

**:HCOPY:PRINTER:FEED**

**Function** Executes paper feeding of the built-in printer. This is an overlap command.

**Syntax** :HCOPY:PRINTER:FEED

**Example** :HCOPY:PRINTER:FEED

**:HCOPY:PRINTER:MAG**

**Function** Sets the magnification for long copy or queries the current setting.

**Syntax** :HCOPY:PRINTER:MAG {<NRf>}  
:HCOPY:PRINTER:MAG?  
<NRf>=2 to 100000000 (see the *DL750 User's Manual*)

**Example** :HCOPY:PRINTER:MAG 10  
:HCOPY:PRINTER:MAG? -> :HCOPY:  
PRINTER:MAG 10.0

**Description** When making a long copy of the waveform measured using the external clock, set the print length using the magnification.

**:HCOPY:PRINTER:RANGE**

**Function** Sets the output range for long copy or queries the current setting.

**Syntax** :HCOPY:PRINTER:RANGE  
{<NRf>}, {<NRf>}  
:HCOPY:PRINTER:RANGE?  
<NRf>=-5.00 to 5.00 (10 div/displayed record length steps)

**Example** :HCOPY:PRINTER:RANGE -5.0,5.0  
:HCOPY:PRINTER:RANGE? -> :HCOPY:  
PRINTER:RANGE -5.0000000,5.0000000

## 6.14 HCOPY Group

---

### **:HCOPY:PRINter:TDIV**

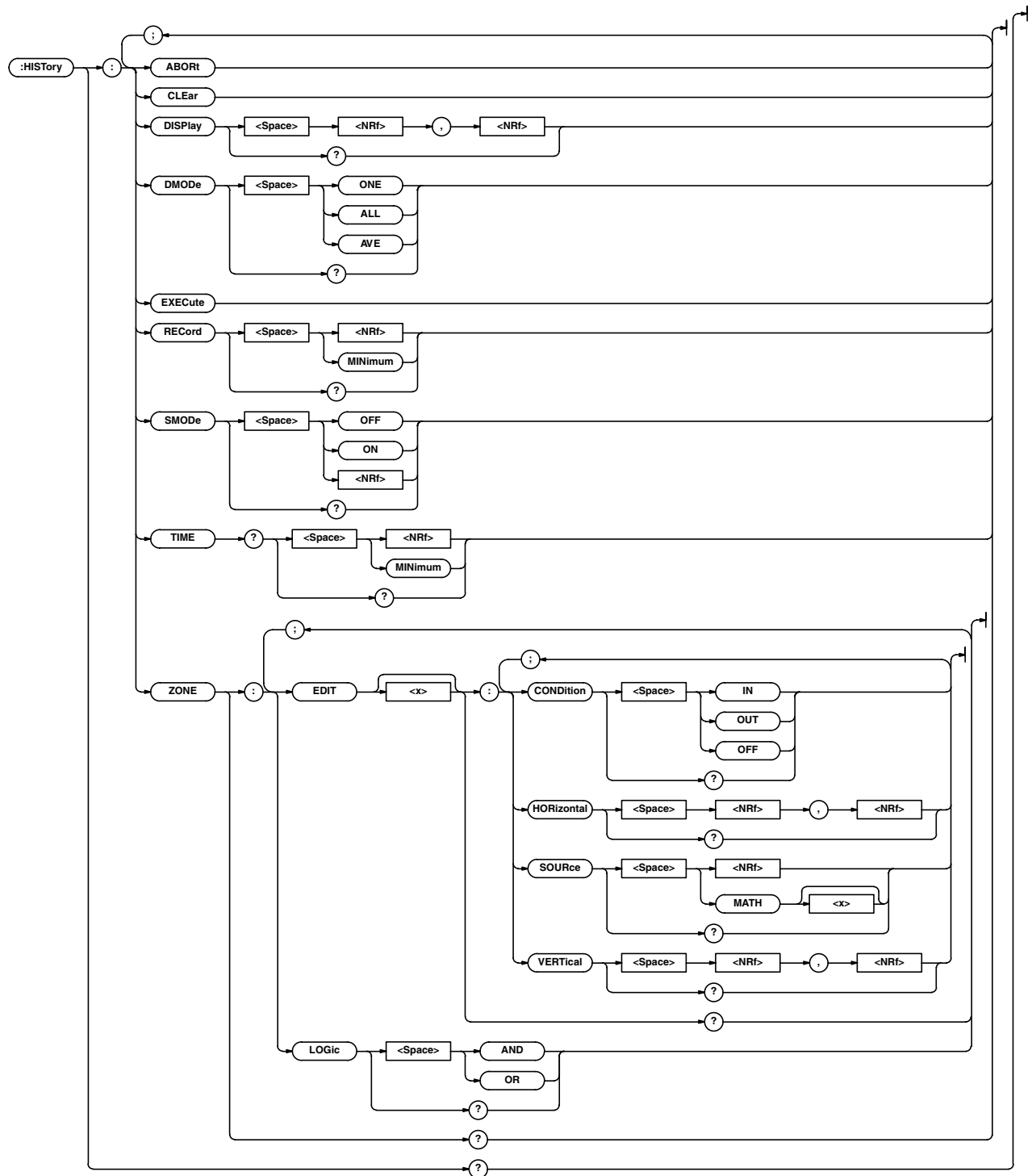
Function	Sets the T/div value for long copy or queries the current setting.
Syntax	:HCOPY:PRINter:TDIV {<Time> HOUR1 HOUR2 HOUR3 HOUR4 HOUR5 HOUR6 HOUR7 HOUR8 HOUR9 HOUR10 HOUR12 DAY1 DAY2 DAY3}
Example	:HCOPY:PRINter:TDIV?<Time>= <i>See the DL750 User's Manual</i> :HCOPY:PRINter:TDIV 200MS :HCOPY:PRINter:TDIV? -> :HCOPY:PRINter:TDIV 200.0E-03
Description	When making a long copy of the waveforms measured using the internal clock, set the print length using a T/div value.

### **:HCOPY:PRINter:TYPE**

Function	Sets normal copy or long copy when outputting to the built-in printer or queries the current setting.
Syntax	:HCOPY:PRINter:TYPE {NORMAL LONG}
Example	:HCOPY:PRINter:TYPE LONG :HCOPY:PRINter:TYPE? -> :HCOPY:PRINter:TYPE LONG

## 6.15 HISTory Group

The commands in this group deal with data loading from the history memory. You can make the same settings, execute the same operations, and make the same inquiries as when the HISTORY key on the front panel is used.



## 6.15 HISToRY Group

### **:HISToRY?**

**Function** Queries all settings related to the history memory function.

**Syntax** :HISToRY?

**Example** :HISToRY? -> :HISToRY:RECORD 0;  
DMODE ONE;DISPLAY 0,0;SMODE 0;ZONE:  
EDIT1:CONDITION OFF;SOURCE 1;  
HORIZONTAL 0.50,-0.50;  
VERTICAL -3.0000000,-2.5000000;:  
HISToRY:ZONE:EDIT2:CONDITION OFF;  
SOURCE 1;HORIZONTAL 0.50,-0.50;  
VERTICAL -1.0000000,-0.5000000;:  
HISToRY:ZONE:EDIT3:CONDITION OFF;  
SOURCE 1;HORIZONTAL 0.50,-0.50;  
VERTICAL 1.0000000,1.5000000;:  
HISToRY:ZONE:EDIT4:CONDITION OFF;  
SOURCE 1;HORIZONTAL 0.50,-0.50;  
VERTICAL3.0000000,3.5000000;:HISToRY:  
ZONE:LOGIC AND

### **:HISToRY:ABORT**

**Function** Aborts the search.

**Syntax** :HISToRY:ABORT

**Example** :HISToRY:ABORT

### **:HISToRY:Clear**

**Function** Clears the history memory data.

**Syntax** :HISToRY:Clear

**Example** :HISToRY:CLEAR

**Description** Executing this command will clear all the waveform data.

### **:HISToRY:DISPlay**

**Function** Sets the start number and end number of the display record or queries the current setting.

**Syntax** :HISToRY:DISPlay {<NRf>,<NRf>}  
:HISToRY:DISPlay?  
<NRf>=0 to -1999

**Example** :HISToRY:DISPlay -2,-10  
:HISToRY:DISPlay? -> :HISToRY:  
DISPlay -2,-10

**Description** The selectable number of records varies depending on the acquisition setting and memory extension.

### **:HISToRY:DMoDe (Display Mode)**

**Function** Sets whether to display a single record or all records accumulated or queries the current setting.

**Syntax** :HISToRY:DMoDe {ONE|ALL|AVE}  
:HISToRY:DMoDe?

**Example** :HISToRY:DMoDe ALL  
:HISToRY:DMoDe? -> :HISToRY:  
DMoDe ALL

**Description** If you specify ALL, the record numbers specified by :HISToRY:DISPlay are displayed accumulated.

### **:HISToRY:EXECute**

**Function** Executes the search. This is an overlap command.

**Syntax** :HISToRY:EXECute

**Example** :HISToRY:EXECUTE

### **:HISToRY:RECORD**

**Function** Sets the target record or queries the current setting.

**Syntax** :HISToRY:RECORD {<NRf>|MINimum}  
:HISToRY:RECORD?  
<NRf>=0 to -1999

**Example** :HISToRY:RECORD 0  
:HISToRY:RECORD? -> :HISToRY:  
RECORD 0

**Description**

- Specifying "MINimum" sets the record to the minimum record number.
- The selectable target records vary depending on the record length and memory extension settings.

### **:HISToRY:RECORD? MINimum**

**Function** Queries the minimum record number.

**Syntax** :HISToRY:RECORD? MINimum

**Example** :HISToRY:RECORD? MINIMUM ->  
:HISToRY:RECORD -538

**Description** The minimum record number varies depending on the record length and memory extension settings.

### **:HISToRY:SMoDe (Search Mode)**

**Function** Sets the history memory search mode or queries the current setting.

**Syntax** :HISToRY:SMoDe {<Boolean>}  
:HISToRY:SMoDe?

**Example** :HISToRY:SMoDe OFF  
:HISToRY:SMoDe? -> :HISToRY:SMoDe 0

### **:HISToRY:TIME?**

**Function** Queries the trigger time of the target record number.

**Syntax** :HISToRY:TIME? {<NRf>|MINimum}  
<NRf>=0 to -1999

**Example** :HISToRY:TIME? -1 -> :HIST:  
TIME "-0001 10:20:30.04"

**Description**

- Specifying "MINimum" sets the record to the minimum record number.
- The selectable target records vary depending on the record length and memory extension settings.

### **:HISToRY:ZONE?**

**Function** Queries all settings related to the zone search.

**Syntax** :HISToRY:ZONE?

**Example** :HISToRY:ZONE? -> :HISToRY:ZONE:  
EDIT1:CONDITION OFF;SOURCE 1;  
HORIZONTAL 0.50,-0.50;  
VERTICAL -3.0000000,-2.5000000;:  
HISToRY:ZONE:EDIT2:CONDITION OFF;  
SOURCE 1;HORIZONTAL 0.50,-0.50;  
VERTICAL -1.0000000,-0.5000000;:  
HISToRY:ZONE:EDIT3:CONDITION OFF;  
SOURCE 1;HORIZONTAL 0.50,-0.50;  
VERTICAL1.0000000,1.5000000;:HISToRY:  
ZONE:EDIT4:CONDITION OFF;SOURCE 1;  
HORIZONTAL 0.50,-0.50;  
VERTICAL3.0000000,3.5000000;:HISToRY:  
ZONE:LOGIC AND

**:HISTory:ZONE:EDIT<x>?**

**Function** Queries all settings related to the specified zone cursor.

**Syntax** :HISTory:ZONE:EDIT<x>?  
<x>=1 to 4

**Example** :HISTORY:ZONE:EDIT1? -> :HISTORY:  
ZONE:EDIT1:CONDITION OFF;SOURCE 1;  
HORIZONTAL 0.50,-0.50;  
VERTICAL -3.0000000,-1.0000000

**:HISTory:ZONE:EDIT<x>:CONDition**

**Function** Sets the search condition of the specified zone cursor or queries the current setting.

**Syntax** :HISTory:ZONE:EDIT<x>:CONDition  
{OFF|IN|OUT}  
:HISTory:ZONE:EDIT<x>:CONDition?  
<x>=1 to 4

**Example** :HISTORY:ZONE:EDIT1:CONDITION IN  
:HISTORY:ZONE:EDIT1:CONDITION? ->  
:HISTORY:ZONE:EDIT1:CONDITION IN

**Description** Use EDIT<x> to specify the desired zone cursor.

**:HISTory:ZONE:EDIT<x>:HORizontal**

**Function** Sets the vertical position (Upper/Lower) of the specified zone cursor or queries the current setting.

**Syntax** :HISTory:ZONE:EDIT<x>:HORizontal  
{<Nrf>,<Nrf>}  
:HISTory:ZONE:EDIT<x>:HORizontal?  
<x>=1 to 4  
<Nrf>=-5 to 5 div (0.01 steps, in the order  
Upper and Lower)

**Example** :HISTORY:ZONE:EDIT1:  
HORIZONTAL 4.00,-4.00  
:HISTORY:ZONE:EDIT1:HORIZONTAL? ->  
:HISTORY:ZONE:EDIT1:  
HORIZONTAL 4.00,-4.00

**Description** Use EDIT<x> to specify the desired zone cursor.

**:HISTory:ZONE:EDIT<x>:SOURCE**

**Function** Sets the target waveform of the specified zone cursor or queries the current setting.

**Syntax** :HISTory:ZONE:EDIT<x>:SOURCE  
{<Nrf>}  
:HISTory:ZONE:EDIT<x>:SOURCE?  
<x> of EDIT<x>=1 to 4  
<Nrf>=1 to 16

**Example** :HISTORY:ZONE:EDIT1:SOURCE 1  
:HISTORY:ZONE:EDIT1:SOURCE? ->  
:HISTORY:ZONE:EDIT1:SOURCE 1

**Description** Use EDIT<x> to specify the desired zone cursor.

**:HISTory:ZONE:EDIT<x>:VERTical**

**Function** Sets the horizontal position (Left/Right) of the specified zone cursor or queries the current setting.

**Syntax** :HISTory:ZONE:EDIT<x>:VERTical  
{<Nrf>,<Nrf>}  
:HISTory:ZONE:EDIT<x>:VERTical?  
<x>=1 to 4  
<Nrf>=See the *DL750 User's Manual* (in the  
order Left and Right)

**Example** :HISTORY:ZONE:EDIT1:VERTICAL -2,2  
:HISTORY:ZONE:EDIT1:VERTICAL? ->  
:HISTORY:ZONE:EDIT1:  
VERTICAL -2.0000000,2.0000000

**Description** Use EDIT<x> to specify the desired zone cursor.

**:HISTory:ZONE:LOGic**

**Function** Sets the search logic condition between the specified zone cursors or queries the current setting.

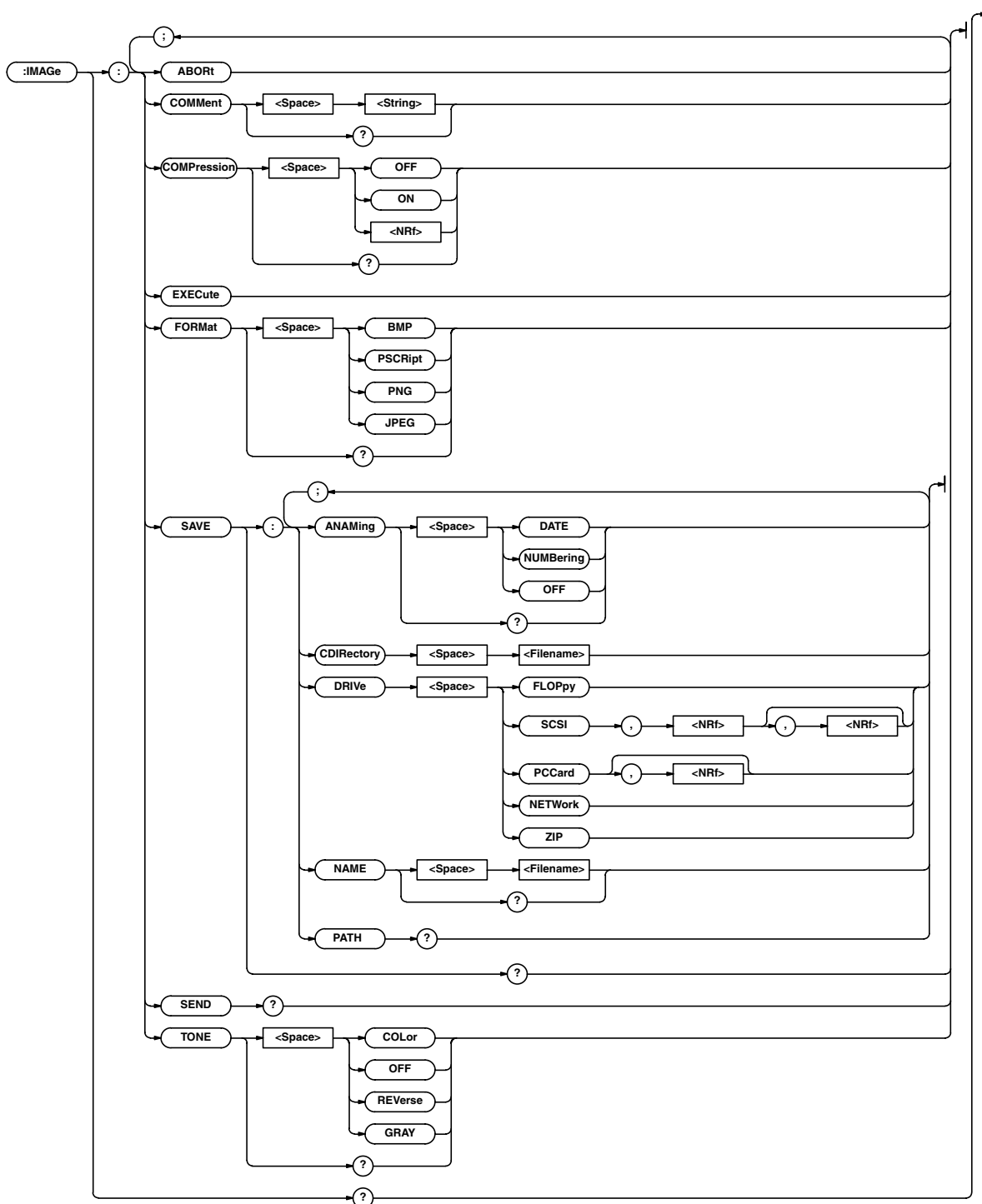
**Syntax** :HISTory:ZONE:LOGic {AND|OR}  
:HISTory:ZONE:LOGic?

**Example** :HISTORY:ZONE:LOGIC AND  
:HISTORY:ZONE:LOGIC? -> :HISTORY:  
ZONE:LOGIC AND



### 6.16 IMAGE Group

The commands in this group deal with the output of screen image data. You can make the same settings, execute the same operations, and make the same inquiries of settings as when (SHIFT+) IMAGE SAVE on the front panel is used.



**: IMAGE?**

Function Queries all settings of the screen image data output.

Syntax :IMAGE?

Example :IMAGE? -> :IMAGE:FORMAT BMP;  
TONE COLOR;COMMENT "" ;SAVE:  
ANAMING DATE;NAME "ABC" ; :IMAGE:  
COMPRESSION 1

**: IMAGE:ABORT**

Function Aborts saving the screen image data to the storage medium.

Syntax :IMAGE:ABORT

Example :IMAGE:ABORT

**: IMAGE:COMMENT**

Function Sets the comment on the lower right corner of the screen or queries the current setting.

Syntax :IMAGE:COMMENT <String>

:IMAGE:COMMENT?

Example :IMAGE:COMMENT "THIS IS TEST."  
:IMAGE:COMMENT? -> :IMAGE:  
COMMENT "THIS IS TEST."

**: IMAGE:COMPRESSION**

Function Enables or disables the data compression of screen data in BMP format or queries the current setting.

Syntax :IMAGE:COMPRESSION {<Boolean>}

:IMAGE:COMPRESSION?

Example :IMAGE:COMPRESSION ON  
:IMAGE:COMPRESSION? -> :IMAGE:  
COMPRESSION 1

Description This setting is valid when saving to the storage medium.

**: IMAGE:EXECUTE**

Function Stores the screen image data to the storage medium.

Syntax :IMAGE:EXECUTE

Example :IMAGE:EXECUTE

**: IMAGE:FORMAT**

Function Sets the output format of the screen image data or queries the current setting.

Syntax :IMAGE:FORMAT

{BMP|PSCRIPT|PNG|JPEG}  
:IMAGE:FORMAT?

Example :IMAGE:FORMAT PNG  
:IMAGE:FORMAT? -> :IMAGE:FORMAT PNG

**: IMAGE:SAVE?**

Function Queries all settings related to the saving of the screen image data to the storage medium.

Syntax :IMAGE:SAVE?

Example :IMAGE:SAVE? -> :IMAGE:SAVE:  
ANAMING DATE;NAME "ABC"

**: IMAGE:SAVE:ANAMING**

Function Sets the auto naming function of output file names or queries the current setting.

Syntax :IMAGE:ANAMING {DATE|NUMBERING|OFF}

:IMAGE:SAVE:ANAMING?

Example :IMAGE:ANAMING DATE  
:IMAGE:SAVE:ANAMING? -> :IMAGE:  
SAVE:ANAMING DATE

**: IMAGE:SAVE:CDIRECTORY**

Function Changes the file directory.

Syntax :IMAGE:SAVE:DIRECTORY <String>

<String>=Up to 16 characters

Example :IMAGE:SAVE:DIRECTORY "NO\_1"

**: IMAGE:SAVE:DRIVE**

Function Sets the storage medium on which to create the file or queries the current setting.

Syntax :IMAGE:SAVE:DRIVE

{FLOPPY|SCSI,<Nrf>,[<Nrf>]|

PCCard,[<Nrf>]|ZIP|NETWork}

:IMAGE:SAVE:DRIVE?

The first <Nrf> of SCSI=SCSI address (0 to 7)

The second <Nrf> of SCSI=Partitions (0 to 2)

<Nrf> of PCCard=Partitions (0 to 1)

Example :IMAGE:SAVE:DRIVE FLOPPY

:IMAGE:SAVE:DRIVE? -> :IMAGE:SAVE:

DRIVE FLOPPY

Description

- If the drive does not contain partitions, omit the <Nrf> corresponding to partitions. "NETWork" can be used when the Ethernet interface (option) is installed.

- The REALTIME recording partition (SCSI address=internal ID, partition=0) cannot be specified.

**: IMAGE:SAVE:NAME**

Function Sets the name of the file to be created or queries the current setting.

Syntax :IMAGE:SAVE:NAME <Filename>

:IMAGE:SAVE:NAME?

Example :IMAGE:SAVE:NAME "DISP\_1"  
:IMAGE:SAVE:NAME? -> :IMAGE:SAVE:  
NAME "DISP\_1"

**: IMAGE:SAVE:PATH?**

Function Queries the current directory.

Syntax :IMAGE:SAVE:PATH?

Example :IMAGE:SAVE:PATH? -> :IMAGE:SAVE:  
PATH "Path=HD-0"

**: IMAGE:SEND?**

Function Queries the screen image data.

Syntax :IMAGE:SEND?

Example :IMAGE:SEND? -> #6  
(number of bytes, 6 digits) (data byte sequence)  
(Block data)

Description

- The number of bytes of <Block data> is {2 + 6 + number of data points + 1K (delimiter)}.
- For details on <Block data>, see page 5-7.
- If you specify :IMAGE:FORMAT PSCRIPT, data is not output. "#6000000" is returned.

**: IMAGE:TONE**

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

Syntax :IMAGE:TONE

{OFF|COLOR|REVERSE|GRAY}  
:IMAGE:TONE?

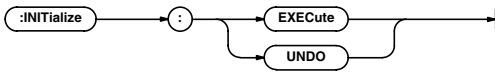
Example :IMAGE:TONE COLOR  
:IMAGE:TONE? -> :IMAGE:TONE COLOR

## 6.17 INITIALize Group

---

### 6.17 INITIALize Group

The commands in this group deal with initialization. You can execute the same operation as Initialize in the SETUP key on the front panel.



#### **:INITialize:EXECute**

Function Executes the initialization.

Syntax :INITialize:EXECute

Example :INITIALIZE:EXECUTE

#### **:INITialize:UNDO**

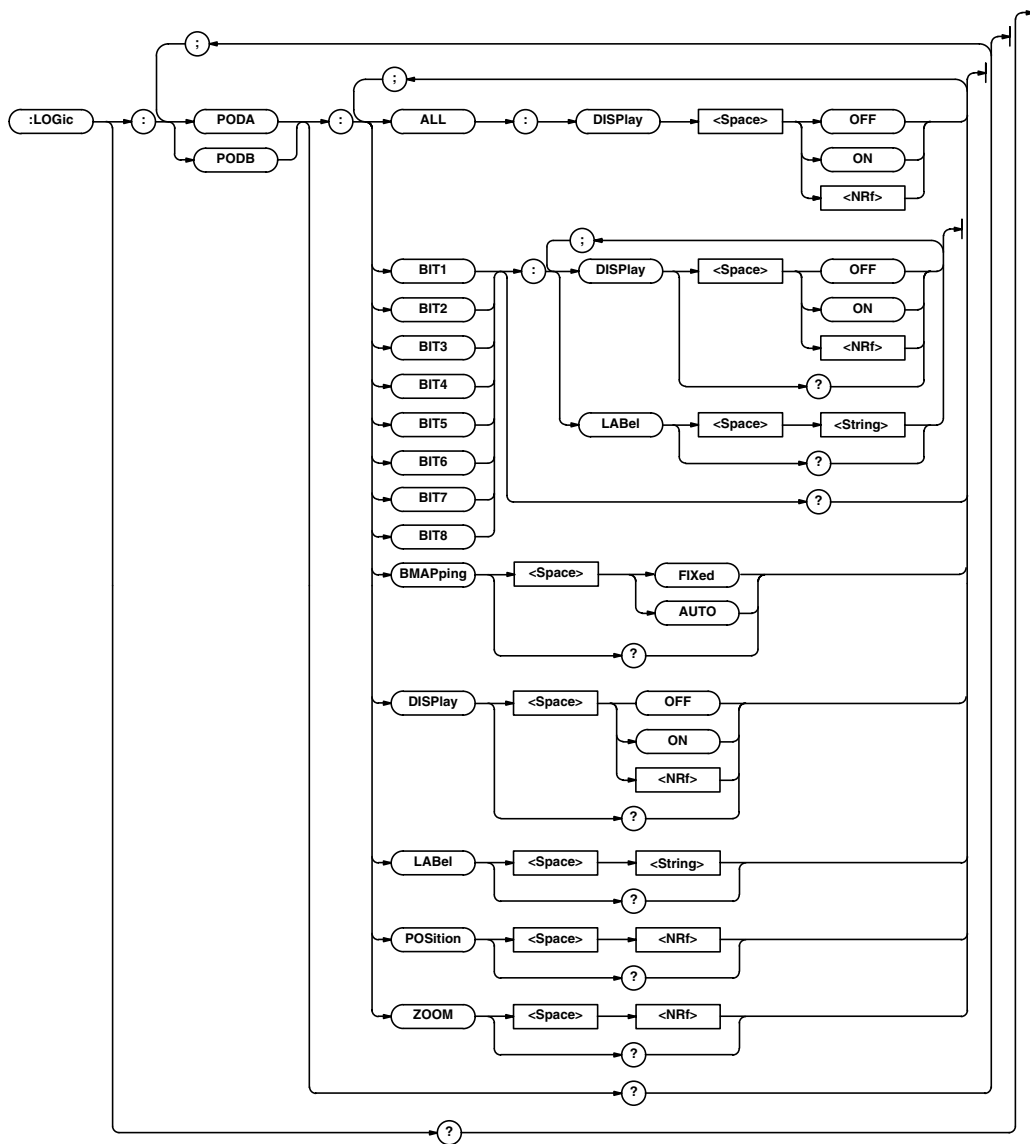
Function Cancels the initialization that was executed.

Syntax :INITialize:UNDO

Example :INITIALIZE:UNDO

### 6.18 LOGic Group

The commands in this group deal with the logic probe. You can make the same settings and inquiries as when the (SHIFT)+CH9 (LOGIC A) key or the (SHIFT)+CH10 (LOGIC B) key on the front panel is used.



## 6.18 LOGic Group

### **:LOGic?**

**Function** Queries all settings related to the logic waveforms.

**Syntax** :LOGic?

**Example**

```
:LOGIC? -> :LOGIC:PODA:
POSITION 0.00;ZOOM 2.500;
LABEL "LogicA";BIT1:DISPLAY 0;
LABEL "A-1";:LOGIC:PODA:BIT2:
DISPLAY 1;LABEL "A-2";:LOGIC:PODA:
BIT3:DISPLAY 1;LABEL "A-3";:LOGIC:
PODA:BIT4:DISPLAY 1;LABEL "A-4";:
LOGIC:PODA:BIT5:DISPLAY 1;
LABEL "A-5";:LOGIC:PODA:BIT6:
DISPLAY 1;LABEL "A-6";:LOGIC:PODA:
BIT7:DISPLAY 1;LABEL "A-7";:LOGIC:
PODA:BIT8:DISPLAY 1;LABEL "A-8";:
LOGIC:PODA:BMAPPING AUTO;:LOGIC:
PODB:POSITION 0.00;ZOOM 1.000;
LABEL "LogicB";BIT1:DISPLAY 1;
LABEL "B-1";:LOGIC:PODB:BIT2:
DISPLAY 1;LABEL "B-2";:LOGIC:PODB:
BIT3:DISPLAY 0;LABEL "B-3";:LOGIC:
PODB:BIT4:DISPLAY 1;LABEL "B-4";:
LOGIC:PODB:BIT5:DISPLAY 1;
LABEL "B-5";:LOGIC:PODB:BIT6:
DISPLAY 0;LABEL "B-6";:LOGIC:PODB:
BIT7:DISPLAY 1;LABEL "B-7";:LOGIC:
PODB:BIT8:DISPLAY 1;LABEL "B-8";:
LOGIC:PODB:BMAPPING AUTO
```

### **:LOGic:{PODA|PODB}**

**Function** Queries all settings related to the specified logic waveform.

**Syntax** :LOGic:{PODA|PODB}?

**Example**

```
:LOGIC:PODA? -> :LOGIC:PODA:
POSITION 0.00;ZOOM 2.500;
LABEL "LogicA";BIT1:DISPLAY 0;
LABEL "A-1";:LOGIC:PODA:BIT2:
DISPLAY 1;LABEL "A-2";:LOGIC:PODA:
BIT3:DISPLAY 1;LABEL "A-3";:LOGIC:
PODA:BIT4:DISPLAY 1;LABEL "A-4";:
LOGIC:PODA:BIT5:DISPLAY 1;
LABEL "A-5";:LOGIC:PODA:BIT6:
DISPLAY 1;LABEL "A-6";:LOGIC:PODA:
BIT7:DISPLAY 1;LABEL "A-7";:LOGIC:
PODA:BIT8:DISPLAY 1;LABEL "A-8";:
LOGIC:PODA:BMAPPING AUTO
```

### **:LOGic:{PODA|PODB}:ALL:DISPlay**

**Function** Turns ON/OFF the display of all bits of the specified logic waveform.

**Syntax** :LOGic:{PODA|PODB}:ALL:DISPlay {<Boolean>}

**Example** :LOGIC:PODA:ALL:DISPLAY ON

### **:LOGic:{PODA|PODB}:{BIT1|BIT2|BIT3|BIT4|BIT5|BIT6|BIT7|BIT8}:DISPlay**

**Function** Turns ON/OFF the display of the bit of the specified logic waveform or queries the current setting.

**Syntax** :LOGic:{PODA|PODB}:{BIT1|BIT2|BIT3|BIT4|BIT5|BIT6|BIT7|BIT8}:DISPlay {<Boolean>}

**Example** :LOGIC:PODA:BIT1:DISPLAY ON  
:LOGIC:PODA:BIT1:DISPLAY? ->  
:LOGIC:PODA:BIT1:DISPLAY 1

### **:LOGic:{PODA|PODB}:{BIT1|BIT2|BIT3|BIT4|BIT5|BIT6|BIT7|BIT8}:LAbel**

**Function** Sets the label of the bit of the specified logic waveform or queries the current setting.

**Syntax** :LOGic:{PODA|PODB}:{BIT1|BIT2|BIT3|BIT4|BIT5|BIT6|BIT7|BIT8}:LAbel <String>

**Example** :LOGIC:PODA:BIT1:LABEL "NO\_1"  
:LOGIC:PODA:BIT1:LABEL? -> :LOGIC:  
PODA:BIT1:LABEL 1

### **:LOGic:{PODA|PODB}:BMAPPING**

**Function** Sets the bit display position of the specified logic waveform or queries the current setting.

**Syntax** :LOGic:{PODA|PODB}:BMAPPING {AUTO|FIXed}

**Example** :LOGIC:PODA:BMAPPING AUTO  
:LOGIC:PODA:BMAPPING? -> :LOGIC:  
PODA:BMAPPING AUTO

### **:LOGic:{PODA|PODB}:DISPlay**

**Function** Turns ON/OFF the display of all bits of the specified logic waveform or queries the current setting.

**Syntax** :LOGic:{PODA|PODB}:DISPlay {<Boolean>}

**Example** :LOGIC:PODA:DISPLAY ON  
:LOGIC:PODA:DISPLAY? -> :LOGIC:  
PODA:DISPLAY 1

### **:LOGic:{PODA|PODB}:LAbel**

**Function** Sets the label of the specified logic waveform or queries the current setting.

**Syntax** :LOGic:{PODA|PODB}:LAbel <String>

**Example** :LOGIC:PODA:LABEL "LABEL\_A"  
:LOGIC:PODA:LABEL? -> :LOGIC:PODA:  
LABEL "LABEL\_A"

**:LOGic:{PODA|PODB}:POSition**

**Function** Sets the vertical position of the specified logic waveform or queries the current setting.

**Syntax** :LOGic:{PODA|PODB}:POSition {<NRf>}  
:LOGic:{PODA|PODB}:POSition?  
<NRf>=-5.00 to 5.00 (div, 0.01 div steps)

**Example** :LOGIC:PODA:POSITION 0.00  
:LOGIC:PODA:POSITION? -> :LOGIC:  
PODA:POSITION 0.00

**:LOGic:{PODA|PODB}:ZOOM**

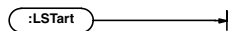
**Function** Sets the vertical zoom rate of the specified logic waveform or queries the current setting.

**Syntax** :LOGic:{PODA|PODB}:ZOOM {<NRf>}  
:LOGic:{PODA|PODB}:ZOOM?  
<NRf>=0.1, 0.111, 0.125, 0.143, 0.167, 0.2, 0.25, 0.33, 0.4, 0.5, 0.556, 0.625, 0.667, 0.714, 0.8, 0.833, 1, 1.11, 1.25, 1.33, 1.43, 1.67, 2, 2.22, 2.5, 3.33, 4, 5, 6.67, 8, 10, 12.5, 16.7, 20, 25, 40, 50, or 100

**Example** :LOGIC:PODA:ZOOM 2.5  
:LOGIC:PODA:ZOOM? -> :LOGIC:PODA:  
ZOOM 2.500

**6.19 LStart Group**

The commands in this group executes log start. Sets the trigger mode to Log and starts the waveform acquisition.

**:LStart** 

**:LStart (Log START)**

**Function** Executes log start.

**Syntax** :LStart

**Example** :LSATART

**:LStart?**

**Function** Executes log start and waits for the completion.

**Syntax** :LStart?

**Example** :LSTART? -> 0

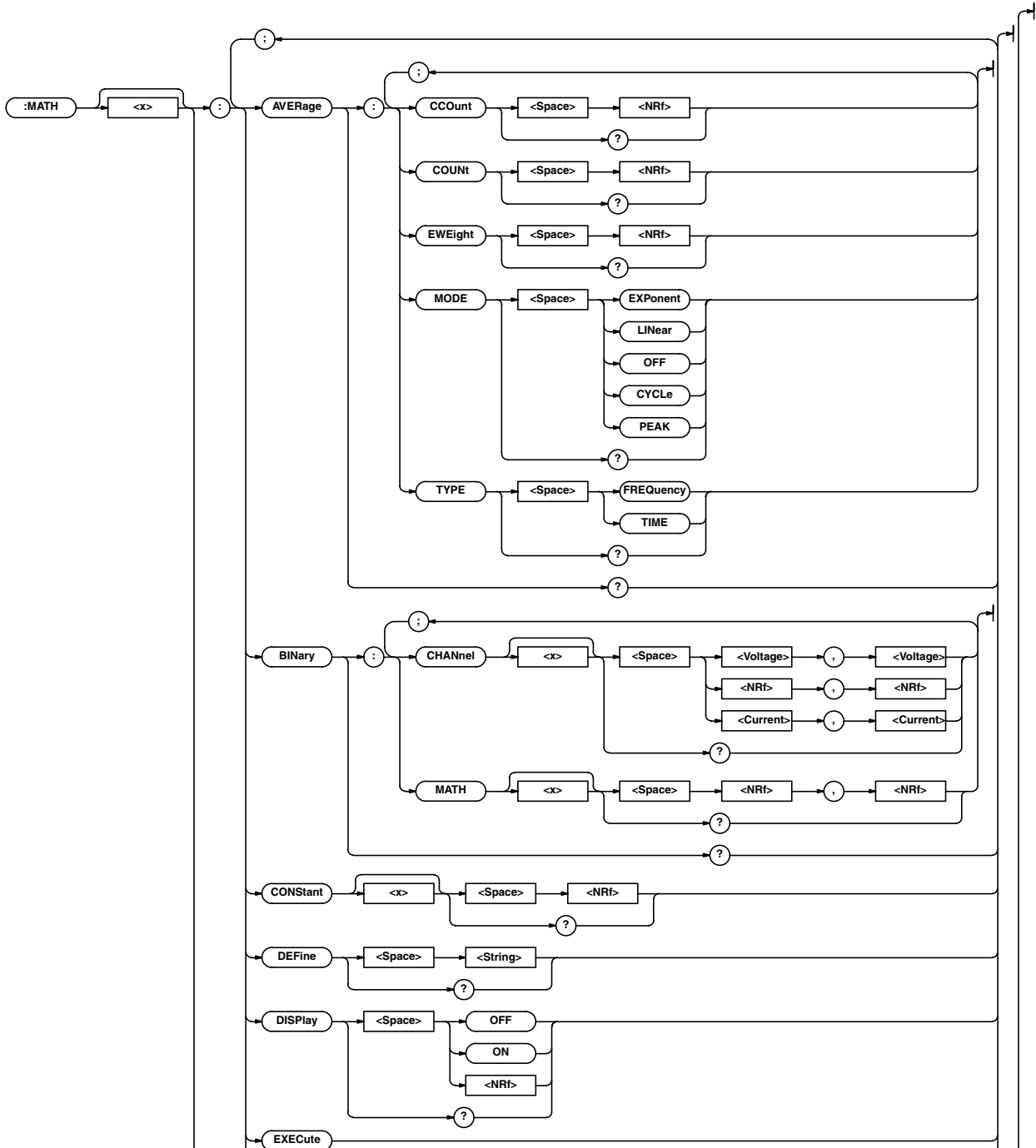
**Description** When the operation is complete, 0 is always returned.

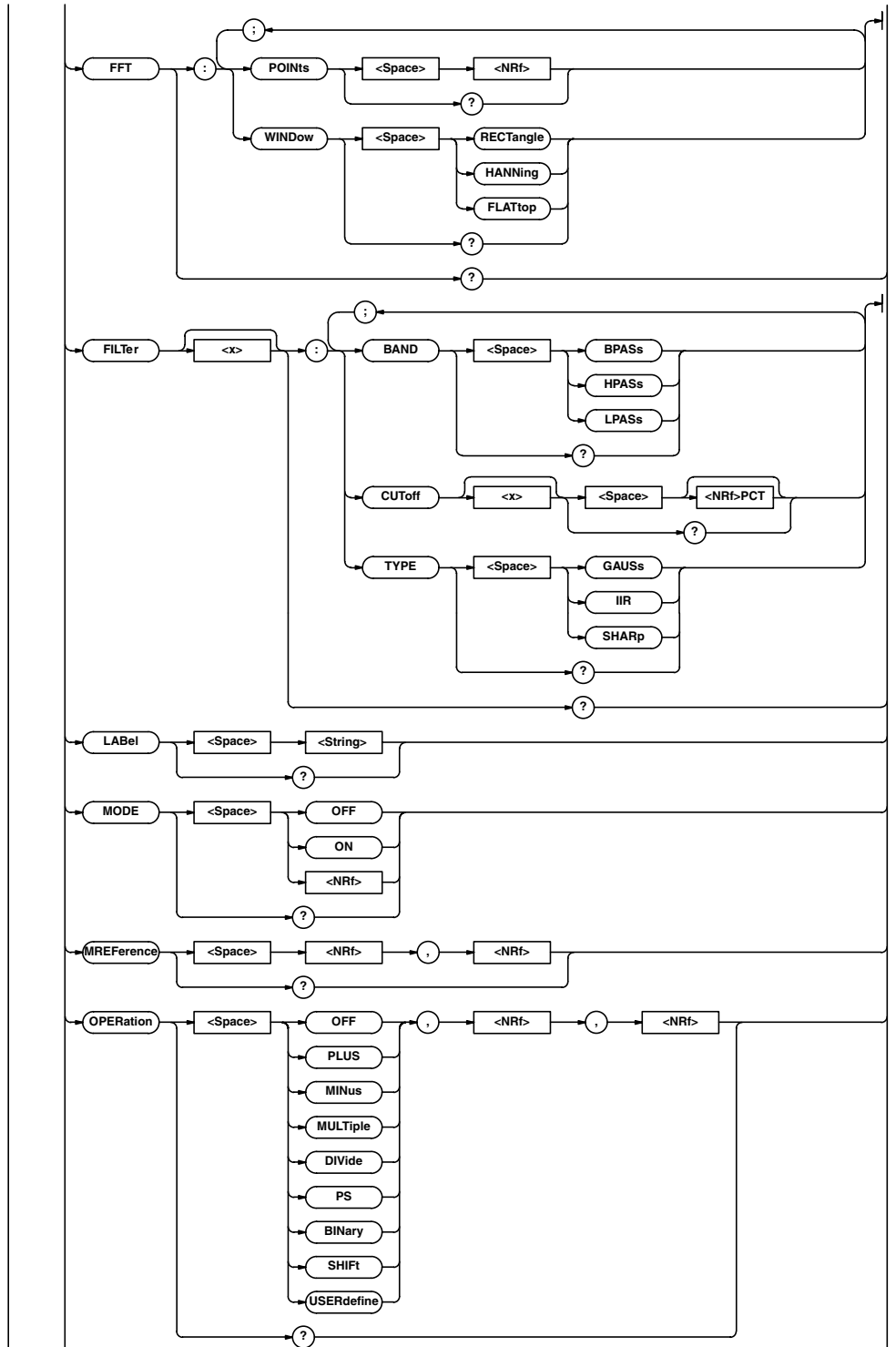
### 6.20 MATH Group

The commands in this group deal with computations. You can make the same settings and inquiries as when the MATH key on the front panel is used.

When the time axis range setting is as follows, the selectable range varies depending on the record length setting. For details, see "Selectable Range of Cursor Position" in section 8.1 of the DL750 User's Manual.

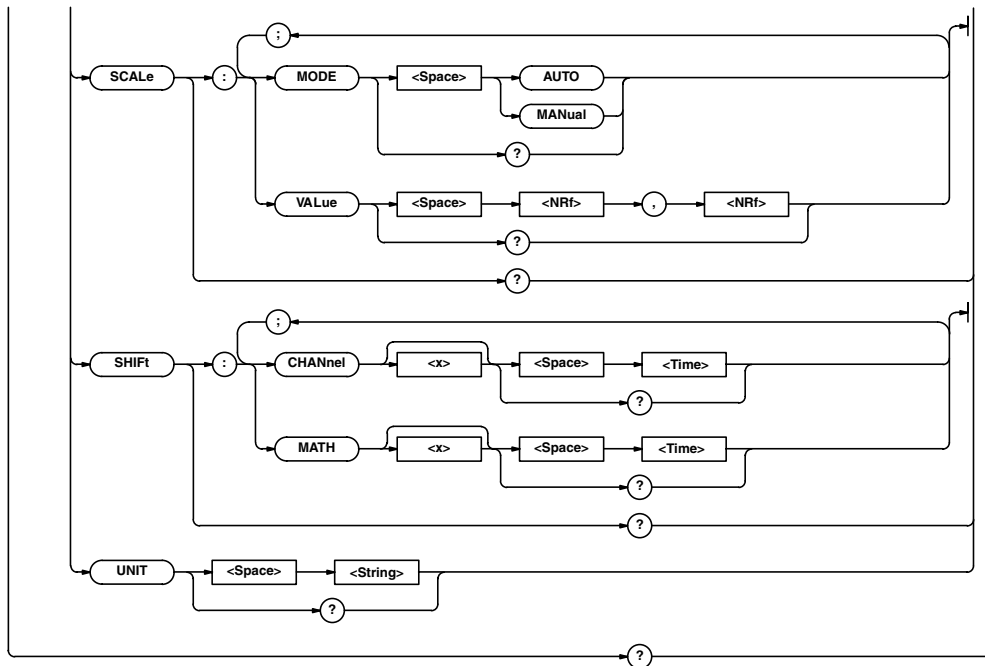
<NRf>=-5 to 5 div







## 6.20 MATH Group



### :MATH<x>?

Function Queries all settings related to the computation.

Syntax :MATH<x>?

<x>=1 to 8

Example :MATH1? -> :MATH1:MODE 1;  
 OPERATION PLUS,1,2;SCALE:MODE AUTO;  
 VALUE 1.00000E+00,-1.00000E+00;;  
 MATH1:UNIT "EU";MATH1:  
 LABEL "MATH1";DISPLAY 1;  
 DEFINE "C1+C2";FFT:POINTS 1000;  
 WINDOW HANNING;;MATH1:  
 MREFERENCE -5.0000000,5.0000000;  
 AVERAGE:MODE LINEAR;TYPE TIME;  
 COUNT 16;EWEIGHT 16;CCOUNT 0;;  
 MATH1:FILTER1:TYPE GAUSS;  
 BAND LPASS;CUTOFF1 10.0;  
 CUTOFF2 10.0;;MATH1:FILTER2:  
 TYPE GAUSS;BAND LPASS;CUTOFF1 10.0;  
 CUTOFF2 10.0;;MATH1:  
 CONSTANT1 1.00000E+00;  
 CONSTANT2 1.00000E+00;  
 CONSTANT3 1.00000E+00;  
 CONSTANT4 1.00000E+00;  
 CONSTANT5 1.00000E+00;  
 CONSTANT6 1.00000E+00;  
 CONSTANT7 1.00000E+00;  
 CONSTANT8 1.00000E+00;SHIFT:  
 CHANNEL1 0.0E+00;CHANNEL2 0.0E+00;  
 CHANNEL3 0.0E+00;CHANNEL4 0.0E+00;  
 CHANNEL5 0.0E+00;CHANNEL6 0.0E+00;  
 CHANNEL7 0.0E+00;CHANNEL8 0.0E+00;  
 CHANNEL9 0.0E+00;CHANNEL10 0.0E+00;  
 CHANNEL11 0.0E+00;  
 CHANNEL12 0.0E+00;  
 CHANNEL13 0.0E+00;  
 CHANNEL14 0.0E+00;  
 CHANNEL15 0.0E+00;  
 CHANNEL16 0.0E+00;  
 CHANNEL17 0.0E+00;  
 CHANNEL18 0.0E+00;  
 CHANNEL19 0.0E+00;  
 CHANNEL20 0.0E+00;

```

CHANNEL21 0.0E+00;
CHANNEL22 0.0E+00;
MATH1 0.0E+00;MATH2 0.0E+00;
MATH3 0.0E+00;MATH4 0.0E+00;
MATH5 0.0E+00;MATH6 0.0E+00;
MATH7 0.0E+00;MATH8 0.0E+00;;MATH1:
BINARY:CHANNEL1 0.0E+00,0.0E+00;
CHANNEL2 0.0E+00,0.0E+00;
CHANNEL3 0.0E+00,0.0E+00;
CHANNEL4 0.0E+00,0.0E+00;
CHANNEL5 0.0E+00,0.0E+00;
CHANNEL6 0.0E+00,0.0E+00
  
```

### :MATH<x>:AVERAGE?

Function Queries all settings related to averaging of the user-defined computation.

Syntax :MATH<x>:AVERAGE?

<x>=1 to 8

Example :MATH1:AVERAGE? -> :MATH1:AVERAGE:  
 MODE LINEAR;TYPE TIME;COUNT 16;  
 EWEIGHT 16;CCOUNT 100

### :MATH<x>:AVERAGE:CCOUNT

Function Sets the cycle count of the cycle average or queries the current setting.

Syntax :MATH<x>:AVERAGE:CCOUNT {<NRf>}  
 :MATH<x>:AVERAGE:CCOUNT?

<x>=1 to 8

<NRf>=10 to 1800

Example :MATH1:AVERAGE:CCOUNT 100  
 :MATH1:AVERAGE:CCOUNT? -> :MATH1:  
 AVERAGE:CCOUNT 100

**:MATH<x>:AVERAge:COUNT**

Function Sets the waveform acquisition count of linear average or queries the current setting.

Syntax :MATH<x>:AVERAge:COUNT {<NRf>}  
:MATH<x>:AVERAge:COUNT?  
<x>=1 to 8  
<NRf>=2 to 128 (2<sup>n</sup> steps)

Example :MATH1:AVERAGE:COUNT 16  
:MATH1:AVERAGE:COUNT? -> :MATH1:  
AVERAGE:COUNT 16

**:MATH<x>:AVERAge:EWEight**

Function Sets the attenuation constant of exponential averaging or queries the current setting.

Syntax :MATH<x>:AVERAge:EWEight {<NRf>}  
:MATH<x>:AVERAge:EWEight?  
<x>=1 to 8  
<NRf>=2 to 256 (2<sup>n</sup> steps)

Example :MATH1:AVERAGE:EWEIGHT 16  
:MATH1:AVERAGE:EWEIGHT? -> :MATH1:  
AVERAGE:EWEIGHT 16

**:MATH<x>:AVERAge:MODE**

Function Sets the averaging mode or queries the current setting.

Syntax :MATH<x>:AVERAge:MODE  
{CYCLe|EXPOnent|LINear|OFF|PEAK}  
:MATH<x>:AVERAge:MODE?  
<x>=1 to 8

Example :MATH1:AVERAGE:MODE LINEAR  
:MATH1:AVERAGE:MODE? -> :MATH1:  
AVERAGE:MODE LINEAR

**:MATH<x>:AVERAge:TYPE**

Function Sets the averaging area or queries the current setting.

Syntax :MATH<x>:AVERAge:TYPE  
{FREQuency|TIME}  
:MATH<x>:AVERAge:TYPE?  
<x>=1 to 8

Example :MATH1:AVERAGE:TYPE TIME  
:MATH1:AVERAGE:TYPE? -> :MATH1:  
AVERAGE:TYPE TIME

**:MATH<x>:BINArY?**

Function Queries all settings related to the binary computation.

Syntax :MATH<x>:BINArY?  
<x>=1 to 8

Example :MATH1:BINARy:  
CHANNEL1 0.0E+00,0.0E+00;  
CHANNEL2 0.0E+00,0.0E+00;  
CHANNEL3 0.0E+00,0.0E+00;  
CHANNEL4 0.0E+00,0.0E+00;  
CHANNEL5 0.0E+00,0.0E+00;  
CHANNEL6 0.0E+00,0.0E+00;  
CHANNEL9 0.0E+00,0.0E+00;  
CHANNEL10 0.0E+00,0.0E+00;  
MATH1 0.00,0.00;MATH2 0.00,0.00;  
MATH3 0.00,0.00;MATH4 0.00,0.00;  
MATH5 0.00,0.00;MATH6 0.00,0.00;  
MATH7 0.00,0.00;MATH8 0.00,0.00

**:MATH<x>:BINArY:{CHANnel<x>|MATH<x>}**

Function Sets the threshold level of the binary computation channel or queries the current setting.

Syntax :MATH<x>:BINArY:{CHANnel<x>|  
MATH1<x>} {<Voltage>,<Current>|  
<Current>,<Current>|<NRf>,<NRf>}  
<x>=1 to 8  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8

For CHANnel<x>: Voltage, <Current>,  
<NRf>=The selectable range varies  
depending on the range and offset settings.  
For details, see the *DL750 User's Manual*.  
For MATH<x>: <NRf>=-5 to 5 div (0.01 div  
steps)

Example :MATH1:BINARy:CHANNEL1 0V,0V  
:MATH1:BINARy:CHANNEL1? -> :MATH1:  
BINARy:CHANNEL1 0.0E+00,0.0E+00

**:MATH<x>:CONStant<x>**

Function Sets the constant of the user-defined computation or queries the current setting.

Syntax :MATH<x>:CONStant<x> {<NRf>}  
:MATH<x>:CONStant<x>?  
<x> of MATH<x>=1 to 8  
<x> of CONStant<x>=1 to 8

<NRf>=-9.9999E+30 to +9.9999E+30

Example :MATH1:CONStANT1 1  
:MATH1:CONStANT1? -> :MATH1:  
CONStANT1 1.00000E+00

**:MATH<x>:DEFine**

Function Sets the equation of the user-defined computation or queries the current setting.

Syntax :MATH<x>:DEFine {<String>}  
:MATH<x>:DEFine?  
<x>=1 to 8  
<String>=Up to 50 characters

Example :MATH1:DEFine "C1-C2"  
:MATH1:DEFine? -> :MATH1:  
DEFine "C1-C2"

Description Only the characters and symbols displayed on the keyboard on the screen can be used. If OPERATION is not USERDEF, the query returns the contents of the OPERATION.

**:MATH<x>:DISPlay**

Function Sets whether to display computed waveforms or queries the current setting.

Syntax :MATH<x>:DISPlay {<Boolean>}  
:MATH<x>:DISPlay?  
<x>=1 to 8

Example :MATH1:DISPlay ON  
:MATH1:DISPlay? -> :MATH1:DISPlay 1

**:MATH<x>:EXECute**

Function Executes the computation.

Syntax :MATH<x>:EXECute  
<x>=1 to 8

Example :MATH1:EXECUTE

## 6.20 MATH Group

### **:MATH<x>:FILTER<x>?**

**Function** Queries all settings related to the filter of the user-defined computation.

**Syntax** :MATH<x>:FILTER<x>?  
<x> of MATH<x>=1 to 8  
<x> of FILTER<x>=1 or 2

**Example** :MATH1:FILTER1? -> :MATH1:FILTER1:  
CUTOFF1 10.0E+00;TYPE GAUSS;  
BAND LPASS

### **:MATH<x>:FILTER<x>:BAND**

**Function** Sets the filter bandwidth or queries the current setting.

**Syntax** :MATH<x>:FILTER<x>:BAND  
{BPASS|HPASS|LPASS}  
:MATH<x>:FILTER<x>:BAND?  
<x> of MATH<x>=1 to 8  
<x> of FILTER<x>=1 or 2

**Example** :MATH1:FILTER1:BAND LPASS  
:MATH1:FILTER1:BAND? -> :MATH1:  
FILTER1:BAND LPASS

**Description** In the case of :MATH<x>:FILTER<x>:TYPE GAUSS, only LPASSs can be specified.

### **:MATH<x>:FILTER<x>:CUTOFF<x>**

**Function** Sets the cutoff frequency or queries the current setting.

**Syntax** :MATH<x>:FILTER<x>:CUTOFF<x>  
{<Nrf>}  
:MATH<x>:FILTER<x>:CUTOFF<x>?  
<x> of MATH<x>=1 to 8  
<x> of FILTER<x>=1 or 2  
<x> of CUTOFF<x>=1 or 2  
<Nrf>=2 to 30% (0.2% steps)

**Example** :MATH1:FILTER1:CUTOFF1 10  
:MATH1:FILTER1:CUTOFF1? -> :MATH1:  
FILTER1:CUTOFF1 10.0

**Description** CUTOFF2 can be specified only for :MATH<x>:FILTER<x>:BAND BPASS.

### **:MATH<x>:FILTER<x>:TYPE**

**Function** Sets the filter type or queries the current setting.

**Syntax** :MATH<x>:FILTER<x>:TYPE  
{GAUSS|IIR|SHARP}  
:MATH<x>:FILTER<x>:TYPE?  
<x>=1 to 8

**Example** :MATH1:FILTER1:TYPE GAUSS  
:MATH1:FILTER1:TYPE? -> :MATH1:  
FILTER1:TYPE GAUSS

### **:MATH<x>:FFT?**

**Function** Queries all settings related to the FFT computation.

**Syntax** :MATH<x>:FFT?  
<x>=1 to 8

**Example** :MATH1:FFT? -> :MATH1:FFT:  
POINTS 1000;WINDOW HANNING

### **:MATH<x>:FFT:POINT**

**Function** Sets the number of points of the FFT computation or queries the current setting.

**Syntax** :MATH<x>:FFT:POINT {<Nrf>}  
:MATH<x>:FFT:POINT?  
<x>=1 to 8  
<Nrf>=1000, 2000, or 10000

**Example** :MATH1:FFT:POINT 1000  
:MATH1:FFT:POINT? -> :MATH1:FFT:  
POINTS 1000

### **:MATH<x>:FFT:WINDOW**

**Function** Sets the window function of the FFT computation or queries the current setting.

**Syntax** :MATH<x>:FFT:WINDOW  
{HANNING|RECTANGLE|FLATTOP}  
:MATH<x>:FFT:WINDOW?  
<x>=1 to 8

**Example** :MATH1:FFT:WINDOW HANNING  
:MATH1:FFT:WINDOW? -> :MATH1:FFT:  
WINDOW HANNING

### **:MATH<x>:LABEL**

**Function** Sets the label of the computation or queries the current setting.

**Syntax** :MATH<x>:LABEL {<String>}  
:MATH<x>:LABEL?  
<x>=1 to 8  
<String>=Up to 8 characters

**Example** :MATH1:LABEL "MATH1"  
:MATH1:LABEL? -> :MATH1:  
LABEL "MATH1"

### **:MATH<x>:MODE**

**Function** Sets the computation mode or queries the current setting.

**Syntax** :MATH<x>:MODE {<Boolean>}  
:MATH<x>:MODE?  
<x>=1 to 8

**Example** :MATH1:MODE ON  
:MATH1:MODE? -> :MATH:MODE 1

### **:MATH<x>:MREFERENCE**

**Function** Sets the computation range or queries the current setting.

**Syntax** :MATH<x>:MREFERENCE {<Nrf>,<Nrf>}  
:MATH<x>:MREFERENCE?  
<x>=1 to 8  
<Nrf>=-5 to 5 divisions (10 divisions/displayed record length steps)

**Example** :MATH1:MREFERENCE -4,4  
:MATH1:MREFERENCE? -> :MATH1:  
MREFERENCE -4.0000000,4.0000000

**:MATH<x>:OPERation**

Function	Sets the computation details or queries the current setting.
Syntax	:MATH<x>:OPERation {PLUS MINus MULtiple BINary DIVide SHIFt USERdefine PS OFF}, {<NRf>} [, {<NRf> MATH1}] :MATH<x>:OPERation? <x>=1 to 8 <NRf>=1 to 4
Example	:MATH1:OPERATION PLUS,1,2 :MATH1:OPERATION? -> :MATH1: OPERATION PLUS,1,2
Description	<ul style="list-style-type: none"> <li>For unary operators (BINary SHIFt PS), select the target waveform using the first &lt;NRf&gt;.</li> <li>For binary operators (PLUS MINus MULtiple DIVide), select the target waveform of the first term using the first &lt;NRf&gt; and the target waveform of the second term using the second &lt;NRf&gt;.</li> </ul>

**:MATH<x>:SCALE?**

Function	Queries all settings related to scaling.
Syntax	:MATH<x>:SCALE? <x>=1 to 8
Example	:MATH1:SCALE? -> :MATH1:SCALE: MODE AUTO;VALUE 200.000E+00,0.0E+00

**:MATH<x>:SCALE:MODE**

Function	Sets the scaling mode or queries the current setting.
Syntax	:MATH<x>:SCALE:MODE {AUTO MANual} :MATH<x>:SCALE:MODE? <x>=1 to 8
Example	:MATH1:SCALE:MODE AUTO :MATH1:SCALE:MODE? -> :MATH1:SCALE: MODE AUTO

**:MATH<x>:SCALE:VALue**

Function	Sets the upper and lower limits for manual scaling or queries the current setting.
Syntax	:MATH<x>:SCALE:VALue {<NRf>,<NRf>} :MATH<x>:SCALE:VALue? <x>=1 to 8 <NRf>=-9.9999E+30 to 9.9999E+30
Example	:MATH1:SCALE:VALUE 1,-1 :MATH1:SCALE:VALUE? -> :MATH1: SCALE:VALUE 1.00000E+00, -1.00000E+00
Description	The larger of the two specified values is automatically set to the upper limit. This command is valid only during manual scaling.

**:MATH<x>:SHIFt:{CHANnel<x>|MATH<x>}**

Function	Sets the number of shift points or queries the current setting.
Syntax	:MATH<x>:SHIFt {CHANnel<x> MATH<x>} {<Time>} :MATH<x>:SHIFt? <x>=1 to 8 <x> of CHANnel<x>=1 to 16 <x> of MATH<x>=1 to 8 <Time>=500 ns to 2592000 s
Example	:MATH1:SHIFT:CHANNEL1 1 :MATH1:SHIFt:CHANNEL1? -> :MATH1: SCALE:VALUE 1.0000E+00,-1.0000E+00

**:MATH<x>:UNIT**

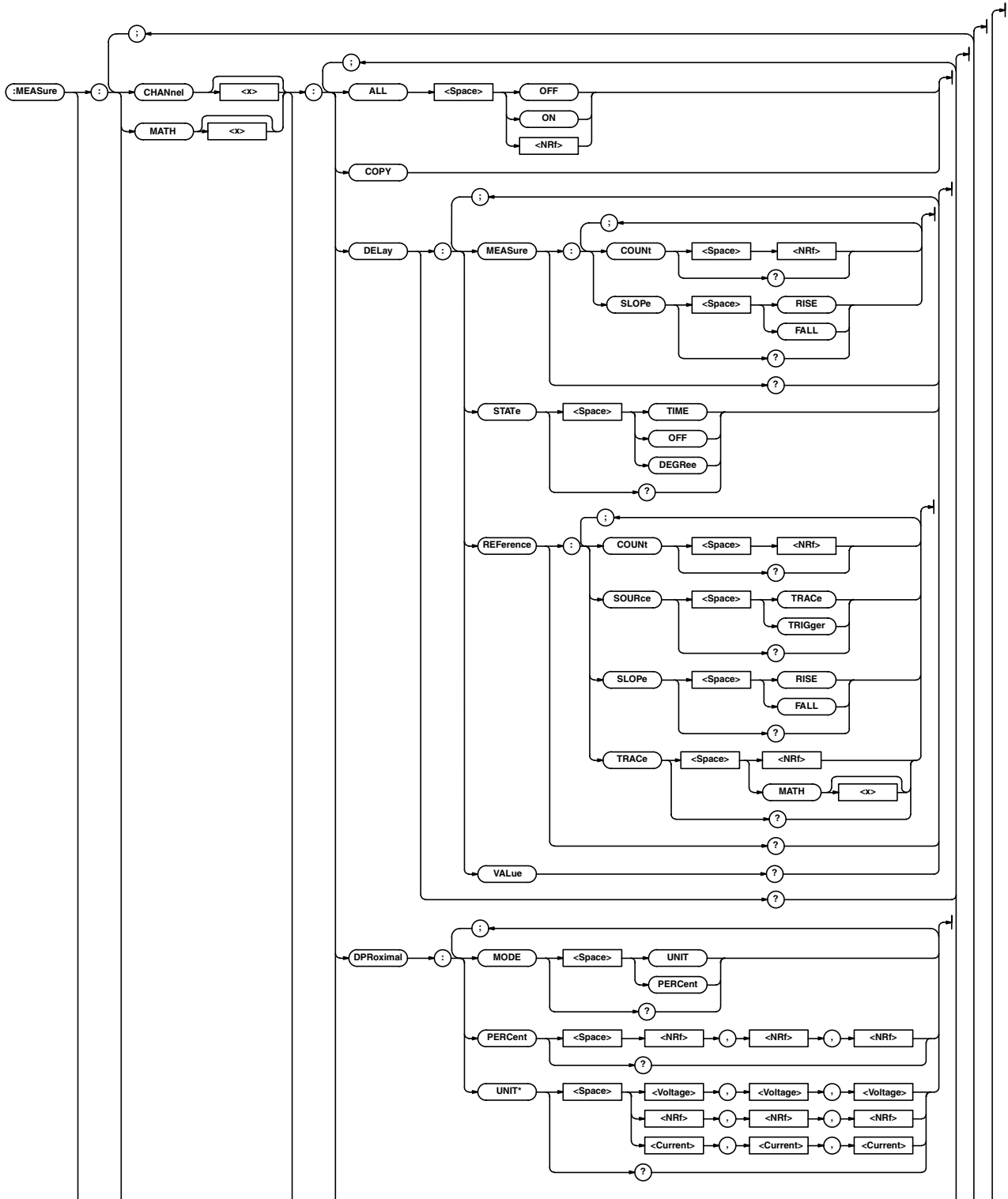
Function	Sets the unit to be added to the computed result or queries the current setting.
Syntax	:MATH<x>:UNIT {<String>} :MATH<x>:UNIT? <x>=1 to 8 <String>=Up to 4 characters
Example	:MATH1:UNIT "EU" :MATH1:UNIT? -> :MATH1:UNIT "EU"
Description	The unit is applied to the scaled value. It does not affect computation results.

### 6.21 MEASure Group

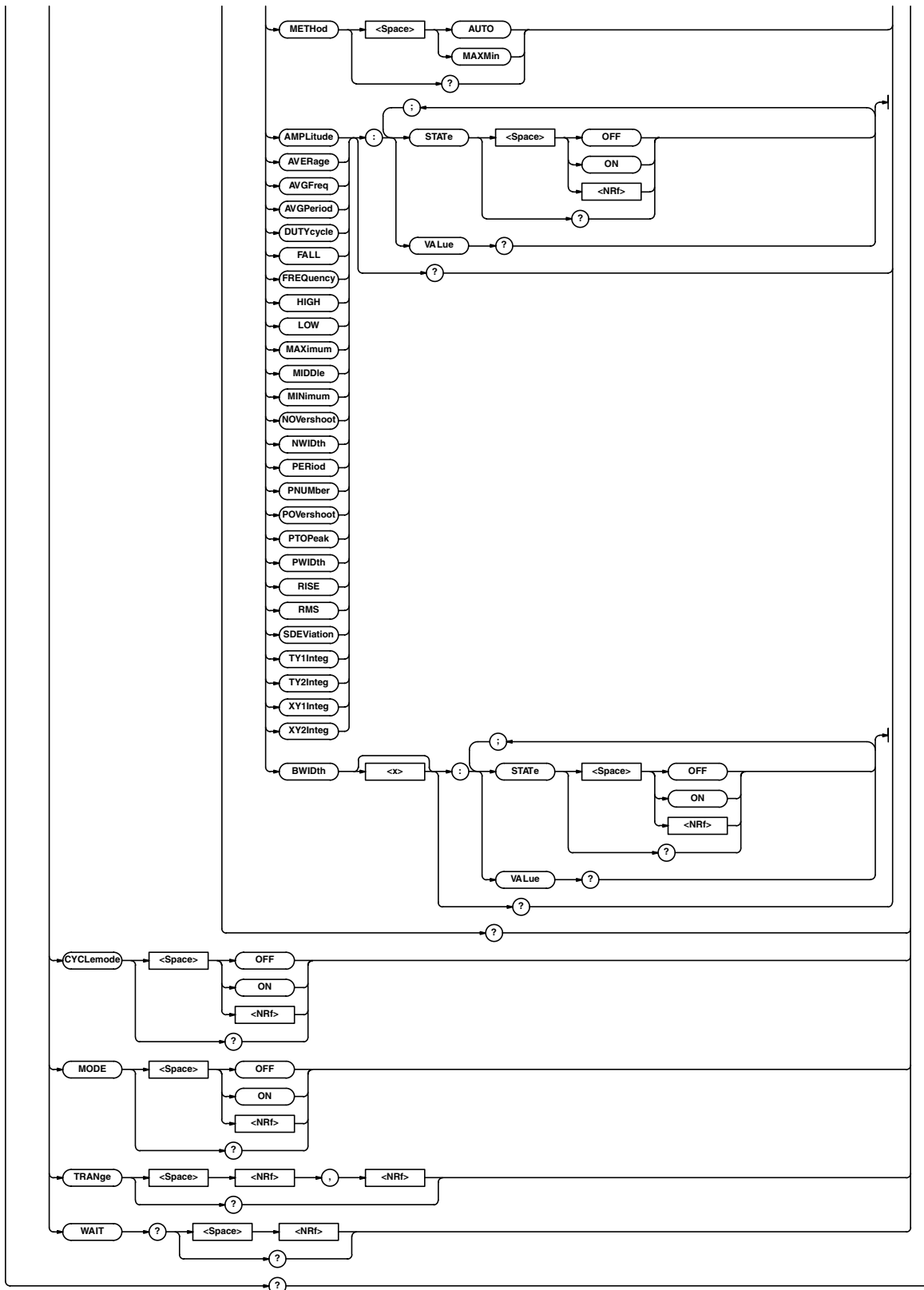
The commands in this group deal with the automated measurement of waveform parameters. You can make the same settings and inquiries of settings and measured values as when the MEASURE key on the front panel is used.

When the time axis range setting is as follows, the selectable range varies depending on the record length setting. For details, see "Selectable Range of Cursor Position" in section 8.1 of the DL750 User's Manual.

<NRf>=-5 to 5 div



\* For :MEASure:MATH<x>, only <NRf> can be specified for the unit.



## 6.21 MEASure Group

### :MEASure?

Function Queries all settings related to the automated measurement of waveform parameters.

Syntax :MEASure?

Example

```
:MEASure? -> :MEASURE:MODE 0;
CHANNEL1:AMPLITUDE:STATE 0;
MEASURE:CHANNEL1:AVERAGE:STATE 0;
MEASURE:CHANNEL1:AVGFREQ:STATE 0;
MEASURE:CHANNEL1:AVGPERIOD:
STATE 0;MEASURE:CHANNEL1:BWIDTH1:
STATE 0;MEASURE:CHANNEL1:BWIDTH2:
STATE 0;MEASURE:CHANNEL1:
DUTYCYCLE:STATE 0;MEASURE:
CHANNEL1:FALL:STATE 0;MEASURE:
CHANNEL1:FREQUENCY:STATE 0;MEASURE:
CHANNEL1:HIGH:STATE 0;MEASURE:
CHANNEL1:LOW:STATE 0;MEASURE:
CHANNEL1:MAXIMUM:STATE 0;MEASURE:
CHANNEL1:MIDDLE:STATE 0;MEASURE:
CHANNEL1:MINIMUM:STATE 0;MEASURE:
CHANNEL1:NOVERSHOOT:STATE 0;
MEASURE:CHANNEL1:NWIDTH:STATE 0;
MEASURE:CHANNEL1:PERIOD:STATE 0;
MEASURE:CHANNEL1:PNUMBER:STATE 0;
MEASURE:CHANNEL1:POVERSHOOT:
STATE 0;MEASURE:CHANNEL1:PTOPEAK:
STATE 0;MEASURE:CHANNEL1:PWIDTH:
STATE 0;MEASURE:CHANNEL1:RISE:
STATE 0;MEASURE:CHANNEL1:RMS:
STATE 0;MEASURE:CHANNEL1:
SDEVIATION:STATE 0;MEASURE:
CHANNEL1:TY1INTEG:STATE 0;
MEASURE:CHANNEL1:TY2INTEG:STATE 0;
MEASURE:CHANNEL1:XY1INTEG:STATE 0;
MEASURE:CHANNEL1:XY2INTEG:STATE 0;
MEASURE:CHANNEL1:DELAY:STATE OFF;
MEASURE:SLOPE RISE;COUNT 1;
MEASURE:CHANNEL1:DELAY:REFERENCE:
SOURCE TRACE;TRACE 1;SLOPE RISE;
COUNT 1;MEASURE:CHANNEL1:
DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT 150.000E+00,0.0E+00,
150.000E+00;MEASURE:CHANNEL1:
METHOD AUTO;MEASURE:CHANNEL2:
AMPLITUDE:STATE 0;MEASURE:
CHANNEL2:AVERAGE:STATE 0;MEASURE:
CHANNEL2:AVGFREQ:STATE 0;MEASURE:
CHANNEL2:AVGPERIOD:STATE 0;
MEASURE:CHANNEL2:BWIDTH1:STATE 0;
MEASURE:CHANNEL2:BWIDTH2:STATE 0;
MEASURE:CHANNEL2:DUTYCYCLE:
STATE 0;MEASURE:CHANNEL2:FALL:
STATE 0;MEASURE:CHANNEL2:
FREQUENCY:STATE 0;MEASURE:
CHANNEL2:HIGH:STATE 0;MEASURE:
CHANNEL2:LOW:STATE 0;MEASURE:
CHANNEL2:MAXIMUM:STATE 0;MEASURE:
CHANNEL2:MIDDLE:STATE 0;MEASURE:
CHANNEL2:MINIMUM:STATE 0;MEASURE:
CHANNEL2:NOVERSHOOT:STATE 0;
MEASURE:CHANNEL2:NWIDTH:STATE 0;
MEASURE:CHANNEL2:PERIOD:STATE 0;
MEASURE:CHANNEL2:PNUMBER:STATE 0;
MEASURE:CHANNEL2:POVERSHOOT:
STATE 0;MEASURE:CHANNEL2:PTOPEAK:
STATE 0;MEASURE:CHANNEL2:PWIDTH:
STATE 0;MEASURE:CHANNEL2:RISE:
```

```
STATE 0;MEASURE:CHANNEL2:RMS:
STATE 0;MEASURE:CHANNEL2:
SDEVIATION:STATE 0;MEASURE:
CHANNEL2:TY1INTEG:STATE 0;MEASURE:
CHANNEL2:TY2INTEG:STATE 0;MEASURE:
CHANNEL2:XY1INTEG:STATE 0;MEASURE:
CHANNEL2:XY2INTEG:STATE 0;MEASURE:
CHANNEL2:DELAY:STATE OFF;MEASURE:
SLOPE RISE;COUNT 1;MEASURE:
CHANNEL2:DELAY:REFERENCE:
SOURCE TRACE;TRACE 2;SLOPE RISE;
COUNT 1;MEASURE:CHANNEL2:
DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -150.000E+00,0.0E+00,
150.000E+00;MEASURE:CHANNEL2:
METHOD AUTO;MEASURE:CHANNEL3:
AMPLITUDE:STATE 0;MEASURE:
CHANNEL3:AVERAGE:STATE 0;MEASURE:
CHANNEL3:AVGFREQ:STATE 0;MEASURE:
CHANNEL3:AVGPERIOD:STATE 0;
MEASURE:CHANNEL3:BWIDTH1:STATE 0;
MEASURE:CHANNEL3:BWIDTH2:STATE 0;
MEASURE:CHANNEL3:DUTYCYCLE:
STATE 0;MEASURE:CHANNEL3:FALL:
STATE 0;MEASURE:CHANNEL3:
FREQUENCY:STATE 0;MEASURE:
CHANNEL3:HIGH:STATE 0;MEASURE:
CHANNEL3:LOW:STATE 0;MEASURE:
CHANNEL3:MAXIMUM:STATE 0;MEASURE:
CHANNEL3:MIDDLE:STATE 0;MEASURE:
CHANNEL3:MINIMUM:STATE 0;MEASURE:
CHANNEL3:NOVERSHOOT:STATE 0;
MEASURE:CHANNEL3:NWIDTH:STATE 0;
MEASURE:CHANNEL3:PERIOD:STATE 0;
MEASURE:CHANNEL3:PNUMBER:STATE 0;
MEASURE:CHANNEL3:POVERSHOOT:
STATE 0;MEASURE:CHANNEL3:PTOPEAK:
STATE 0;MEASURE:CHANNEL3:PWIDTH:
STATE 0;MEASURE:CHANNEL3:RISE:
STATE 0;MEASURE:CHANNEL3:RMS:
STATE 0;MEASURE:CHANNEL3:
SDEVIATION:STATE 0;MEASURE:
CHANNEL3:TY1INTEG:STATE 0;
MEASURE:CHANNEL3:TY2INTEG:STATE 0;
MEASURE:CHANNEL3:XY1INTEG:STATE 0;
MEASURE:CHANNEL3:XY2INTEG:STATE 0;
MEASURE:CHANNEL3:DELAY:STATE OFF;
MEASURE:SLOPE RISE;COUNT 1;
MEASURE:CHANNEL3:DELAY:REFERENCE:
SOURCE TRACE;TRACE 3;SLOPE RISE;
COUNT 1;MEASURE:CHANNEL3:
DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -150.000E+00,0.0E+00,
150.000E+00;MEASURE:CHANNEL3:
METHOD AUTO;MEASURE:CHANNEL4:
AMPLITUDE:STATE 0;MEASURE:
CHANNEL4:AVERAGE:STATE 0;MEASURE:
CHANNEL4:AVGFREQ:STATE 0;MEASURE:
CHANNEL4:AVGPERIOD:STATE 0;
MEASURE:CHANNEL4:BWIDTH1:STATE 0;
MEASURE:CHANNEL4:BWIDTH2:STATE 0;
MEASURE:CHANNEL4:DUTYCYCLE:
STATE 0;MEASURE:CHANNEL4:
FALL:STATE 0;MEASURE:CHANNEL4:
FREQUENCY:STATE 0;MEASURE:
CHANNEL4:HIGH:STATE 0;MEASURE:
```

```

CHANNEL4:LOW:STATE 0;:MEASURE:
CHANNEL4:MAXIMUM:STATE 0;:MEASURE:
CHANNEL4:MIDDLE:STATE 0;:MEASURE:
CHANNEL4:MINIMUM:STATE 0;:MEASURE:
CHANNEL4:NOVERSHOOT:STATE 0;:
MEASURE:CHANNEL4:NWIDTH:STATE 0;:
MEASURE:CHANNEL4:PERIOD:STATE 0;:
MEASURE:CHANNEL4:PNUMBER:STATE 0;:
MEASURE:CANNEL4:POVERSHOOT:
STATE 0;:MEASURE:CHANNEL4:PTOPEAK:
STATE 0;:MEASURE:CHANNEL4:PWIDTh:
STATE 0;:MEASURE:CHANNEL4:RISE:
STATE 0;:MEASURE:CHANNEL4:RMS:
STATE 0;:MEASURE:CHANNEL4:
SDEVIATION:STATE 0;:MEASURE:
CHANNEL4:TY1INTEG:STATE 0;:MEASURE:
CHANNEL4:TY2INTEG:STATE 0;:MEASURE:
CHANNEL4:XY1INTEG:STATE 0;:MEASURE:
CHANNEL4:XY2INTEG:STATE 0;:MEASURE:
CHANNEL4:DELAY:STATE OFF;MEASURE:
SLOPE RISE;COUNT 1;:MEASURE:
CHANNEL4:DELAY:REFERENCE:
SOURCE TRACE;TRACE 4;SLOPE RISE;
COUNT 1;:MEASURE:CHANNEL4:
DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -150.000E+00,0.0E+00,
150.000E+00;:MEASURE:CHANNEL4:
METHOD AUTO;:MEASURE:CHANNEL5:
AMPLITUDE:STATE 0;:MEASURE:
CHANNEL5:AVERAGE:STATE 0;:MEASURE:
CHANNEL5:AVGFREQ:STATE 0;:MEASURE:
CHANNEL5:AVGPERIOD:STATE 0;:
MEASURE:CHANNEL5:BWIDTh1:STATE 0;:
MEASURE:CHANNEL5:BWIDTh2:STATE 0;:
MEASURE:CHANNEL5:DUTYCYCLE:
STATE 0;:MEASURE:CHANNEL5:FALL:
STATE 0;:MEASURE:CHANNEL5:
FREQUENCY:STATE 0;:MEASURE:
CHANNEL5:HIGh:STATE 0;:MEASURE:
CHANNEL5:LOW:STATE 0;:MEASURE:
CHANNEL5:MAXIMUM:STATE 0;:MEASURE:
CHANNEL5:MIDDLE:STATE 0;:MEASURE:
CHANNEL5:MINIMUM:STATE 0;:MEASURE:
CHANNEL5:NOVERSHOOT:STATE 0;:
MEASURE:CHANNEL5:NWIDTH:STATE 0;:
MEASURE:CHANNEL5:PERIOD:STATE 0;:
MEASURE:CHANNEL5:PNUMBER:STATE 0;:
MEASURE:CHANNEL5:POVERSHOOT:
STATE 0;:MEASURE:CHANNEL5:PTOPEAK:
STATE 0;:MEASURE:CHANNEL5:PWIDTh:
STATE 0;:MEASURE:CHANNEL5:RISE:
STATE 0;:MEASURE:CHANNEL5:RMS:
STATE 0;:MEASURE:CHANNEL5:
SDEVIATION:STATE 0;:MEASURE:
CHANNEL5:TY1INTEG:STATE 0;:MEASURE:
CHANNEL5:TY2INTEG:STATE 0;:MEASURE:
CHANNEL5:XY1INTEG:STATE 0;:MEASURE:
CHANNEL5:XY2INTEG:STATE 0;:MEASURE:
CHANNEL5:DELAY:STATE OFF;MEASURE:
SLOPE RISE;COUNT 1;:MEASURE:
CHANNEL5:DELAY:REFERENCE:
SOURCE TRACE;TRACE 5;SLOPE RISE;
COUNT 1;:MEASURE:CHANNEL5:
DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -150.000E+00,0.0E+00,
150.000E+00;:MEASURE:CHANNEL5:

```

```

METHOD AUTO;:MEASURE:CHANNEL6:
AMPLITUDE:STATE 0;:MEASURE:
CHANNEL6:AVERAGE:STATE 0;:MEASURE:
CHANNEL6:AVGFREQ:STATE 0;:MEASURE:
CHANNEL6:AVGPERIOD:STATE 0;:
MEASURE:CHANNEL6:BWIDTh1:STATE 0;:
MEASURE:CHANNEL6:BWIDTh2:STATE 0;:
MEASURE:CHANNEL6:DUTYCYCLE:
STATE 0;:MEASURE:CHANNEL6:FALL:
STATE 0;:MEASURE:CHANNEL6:
FREQUENCY:STATE 0;:MEASURE:
CHANNEL6:HIGh:STATE 0;:MEASURE:
CHANNEL6:LOW:STATE 0;:MEASURE:
CHANNEL6:MAXIMUM:STATE 0;:MEASURE:
CHANNEL6:MIDDLE:STATE 0;:MEASURE:
CHANNEL6:MINIMUM:STATE 0;:MEASURE:
CHANNEL6:NOVERSHOOT:STATE 0;:
MEASURE:CHANNEL6:NWIDTH:STATE 0;:
MEASURE:CHANNEL6:PERIOD:STATE 0;:
MEASURE:CHANNEL6:PNUMBER:STATE 0;:
MEASURE:CHANNEL6:POVERSHOOT:
STATE 0;:MEASURE:CHANNEL6:PTOPEAK:
STATE 0;:MEASURE:CHANNEL6:PWIDTh:
STATE 0;:MEASURE:CHANNEL6:RISE:
STATE 0;:MEASURE:CHANNEL6:RMS:
STATE 0;:MEASURE:
CHANNEL6SDEVIATION:STATE 0;:
MEASURE:CHANNEL6:TY1INTEG:STATE 0;:
MEASURE:CHANNEL6:TY2INTEG:STATE 0;:
MEASURE:CHANNEL6:XY1INTEG:STATE 0;:
MEASURE:CHANNEL6:XY2INTEG:STATE 0;:
MEASURE:CHANNEL6:DELAY:STATE OFF;
MEASURE:SLOPE RISE;COUNT 1;:
MEASURE:CHANNEL6:DELAY:REFERENCE:
SOURCE TRACE;TRACE 6;SLOPE RISE;
COUNT 1;:MEASURE:CHANNEL6:
DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -150.000E+00,0.0E+00,
150.000E+00;:MEASURE:CHANNEL6:
METHOD AUTO;:MEASURE:CHANNEL7:
AMPLITUDE:STATE 0;:MEASURE:
CHANNEL7:AVERAGE:STATE 0;:MEASURE:
CHANNEL7:AVGFREQ:STATE 0;:MEASURE:
CHANNEL7:AVGPERIOD:STATE 0;:
MEASURE:CHANNEL7:BWIDTh1:STATE 0;:
MEASURE:CHANNEL7:BWIDTh2:STATE 0;:
MEASURE:CHANNEL7:DUTYCYCLE:
STATE 0;:MEASURE:CHANNEL7:FALL:
STATE 0;:MEASURE:CHANNEL7:
FREQUENCY:STATE 0;:MEASURE:
CHANNEL7:HIGh:STATE 0;:MEASURE:
CHANNEL7:LOW:STATE 0;:MEASURE:
CHANNEL7:MAXIMUM:STATE 0;:MEASURE:
CHANNEL7:MIDDLE:STATE 0;:MEASURE:
CHANNEL7:MINIMUM:STATE 0;:MEASURE:
CHANNEL7:NOVERSHOOT:STATE 0;:
MEASURE:CHANNEL7:NWIDTH:STATE 0;:
MEASURE:CHANNEL7:PERIOD:STATE 0;:
MEASURE:CHANNEL7:PNUMBER:STATE 0;:
MEASURE:CHANNEL7:POVERSHOOT:
STATE 0;:MEASURE:CHANNEL7:PTOPEAK:
STATE 0;:MEASURE:CHANNEL7:PWIDTh:
STATE 0;:MEASURE:CHANNEL7:RISE:
STATE 0;:MEASURE:CHANNEL7:RMS:
STATE 0;:MEASURE:CHANNEL7:
SDEVIATION:STATE 0;:MEASURE:
CHANNEL7:TY1INTEG:STATE 0;:

```



## 6.21 MEASure Group

MEASURE:CHANNEL7:TY2INTEG:STATE 0;;  
MEASURE:CHANNEL7:XY1INTEG:STATE 0;;  
MEASURE:CHANNEL7:XY2INTEG:STATE 0;;  
MEASURE:CHANNEL7:DELAY:STATE OFF;  
MEASURE:SLOPE RISE;COUNT 1;;  
MEASURE:CHANNEL7:DELAY:REFERENCE:  
SOURCE TRACE;TRACE 7;SLOPE RISE;  
COUNT 1;;MEASURE:CHANNEL7:  
DPROXIMAL:MODE PERCENT;  
PERCENT 10.0,50.0,90.0;  
UNIT -150.000E+00,0.0E+00,  
150.000E+00;;MEASURE:CHANNEL7:  
METHOD AUTO;;MEASURE:CHANNEL8:  
AMPLITUDE:STATE 0;;MEASURE:  
CHANNEL8:AVERAGE:STATE 0;;MEASURE:  
CHANNEL8:AVGFREQ:STATE 0;;MEASURE:  
CHANNEL8:AVGPERIOD:STATE 0;;  
MEASURE:CHANNEL8:BWIDTH1:STATE 0;;  
MEASURE:CHANNEL8:BWIDTH2:STATE 0;;  
MEASURE:CHANNEL8:DUTYCYCLE:  
STATE 0;;MEASURE:CHANNEL8:FALL:  
STATE 0;;MEASURE:CHANNEL8:  
FREQUENCY:STATE 0;;MEASURE:  
CHANNEL8:HIGH:STATE 0;;MEASURE:  
CHANNEL8:LOW:STATE 0;;MEASURE:  
CHANNEL8:MAXIMUM:STATE 0;;MEASURE:  
CHANNEL8:MIDDLE:STATE 0;;MEASURE:  
CHANNEL8:MINIMUM:STATE 0;;MEASURE:  
CHANNEL8:NOVERSHOOT:STATE 0;;  
MEASURE:CHANNEL8:NWIDTH:STATE 0;;  
MEASURE:CHANNEL8:PERIOD:STATE 0;;  
MEASURE:CHANNEL8:PNUMBER:STATE 0;;  
MEASURE:CHANNEL8:POVERSHOOT:  
STATE 0;;MEASURE:CHANNEL8:PTOPEAK:  
STATE 0;;MEASURE:CHANNEL8:PWIDTH:  
STATE 0;;MEASURE:CHANNEL8:RISE:  
STATE 0;;MEASURE:CHANNEL8:RMS:  
STATE 0;;MEASURE:CHANNEL8:  
SDEVIATION:STATE 0;;MEASURE:  
CHANNEL8:TY1INTEG:STATE 0;;MEASURE:  
CHANNEL8:TY2INTEG:STATE 0;;MEASURE:  
CHANNEL8:XY1INTEG:STATE 0;;MEASURE:  
CHANNEL8:XY2INTEG:STATE 0;;MEASURE:  
CHANNEL8:DELAY:STATE OFF;MEASURE:  
SLOPE RISE;COUNT 1;;MEASURE:  
CHANNEL8:DELAY:REFERENCE:  
SOURCE TRACE;TRACE 8;SLOPE RISE;  
COUNT 1;;MEASURE:CHANNEL8:  
DPROXIMAL:MODE PERCENT;  
PERCENT 10.0,50.0,90.0;  
UNIT -150.000E+00,0.0E+00,  
150.000E+00;;MEASURE:CHANNEL8:  
METHOD AUTO;;MEASURE:CHANNEL9:  
AMPLITUDE:STATE 0;;MEASURE:  
CHANNEL9:AVERAGE:STATE 0;;MEASURE:  
CHANNEL9:AVGFREQ:STATE 0;;MEASURE:  
CHANNEL9:AVGPERIOD:STATE 0;;  
MEASURE:CHANNEL9:BWIDTH1:STATE 0;;  
MEASURE:CHANNEL9:BWIDTH2:STATE 0;;  
MEASURE:CHANNEL9:DUTYCYCLE:  
STATE 0;;MEASURE:CHANNEL9:FALL:  
STATE 0;;MEASURE:CHANNEL9:  
FREQUENCY:STATE 0;;MEASURE:  
CHANNEL9:HIGH:STATE 0;;MEASURE:  
CHANNEL9:LOW:STATE 0;;MEASURE:  
CHANNEL9:MAXIMUM:STATE 0;;MEASURE:  
CHANNEL9:MIDDLE:STATE 0;;MEASURE:  
CHANNEL9:MINIMUM:STATE 0;;MEASURE:

CHANNEL9:NOVERSHOOT:STATE 0;;  
MEASURE:CHANNEL9:NWIDTH:STATE 0;;  
MEASURE:CHANNEL9:PERIOD:STATE 0;;  
MEASURE:CHANNEL9:PNUMBER:STATE 0;;  
MEASURE:CHANNEL9:POVERSHOOT:  
STATE 0;;MEASURE:CHANNEL9:PTOPEAK:  
STATE 0;;MEASURE:CHANNEL9:PWIDTH:  
STATE 0;;MEASURE:CHANNEL9:RISE:  
STATE 0;;MEASURE:CHANNEL9:RMS:  
STATE 0;;MEASURE:CHANNEL9:  
SDEVIATION:STATE 0;;MEASURE:  
CHANNEL9:TY1INTEG:STATE 0;;MEASURE:  
CHANNEL9:TY2INTEG:STATE 0;;MEASURE:  
CHANNEL9:XY1INTEG:STATE 0;;MEASURE:  
CHANNEL9:XY2INTEG:STATE 0;;MEASURE:  
CHANNEL9:DELAY:STATE OFF;MEASURE:  
SLOPE RISE;COUNT 1;;MEASURE:  
CHANNEL9:DELAY:REFERENCE:  
SOURCE TRACE;TRACE 9;SLOPE RISE;  
COUNT 1;;MEASURE:CHANNEL9:  
DPROXIMAL:MODE PERCENT;  
PERCENT 10.0,50.0,90.0;  
UNIT -150.000E+00,0.0E+00,  
150.000E+00;;MEASURE:CHANNEL9:  
METHOD AUTO;;MEASURE:CHANNEL10:  
AMPLITUDE:STATE 0;;MEASURE:  
CHANNEL10:AVERAGE:STATE 0;;MEASURE:  
CHANNEL10:AVGFREQ:STATE 0;;MEASURE:  
CHANNEL10:AVGPERIOD:STATE 0;;  
MEASURE:CHANNEL10:BWIDTH1:STATE 0;;  
MEASURE:CHANNEL10:BWIDTH2:STATE 0;;  
MEASURE:CHANNEL10:DUTYCYCLE:  
STATE 0;;MEASURE:CHANNEL10:FALL:  
STATE 0;;MEASURE:CHANNEL10:  
FREQUENCY:STATE 0;;MEASURE:  
CHANNEL10:HIGH:STATE 0;;MEASURE:  
CHANNEL10:LOW:STATE 0;;MEASURE:  
CHANNEL10:MAXIMUM:STATE 0;;MEASURE:  
CHANNEL10:MIDDLE:STATE 0;;MEASURE:  
CHANNEL10:MINIMUM:STATE 0;;MEASURE:  
CHANNEL10:NOVERSHOOT:STATE 0;;  
MEASURE:CHANNEL10:NWIDTH:STATE 0;;  
MEASURE:CHANNEL10:PERIOD:STATE 0;;  
MEASURE:CHANNEL10:PNUMBER:STATE 0;;  
MEASURE:CHANNEL10:POVERSHOOT:  
STATE 0;;MEASURE:CHANNEL10:PTOPEAK:  
STATE 0;;MEASURE:CHANNEL10:PWIDTH:  
STATE 0;;MEASURE:CHANNEL10:RISE:  
STATE 0;;MEASURE:CHANNEL10:RMS:  
STATE 0;;MEASURE:CHANNEL10:  
SDEVIATION:STATE 0;;MEASURE:  
CHANNEL10:TY1INTEG:STATE 0;;  
MEASURE:CHANNEL10:TY2INTEG:  
STATE 0;;MEASURE:CHANNEL10:  
XY1INTEG:STATE 0;;MEASURE:  
CHANNEL10:XY2INTEG:STATE 0;;  
MEASURE:CHANNEL10:DELAY:STATE OFF;  
MEASURE:SLOPE RISE;COUNT 1;;  
MEASURE:CHANNEL10:DELAY:REFERENCE:  
SOURCE TRACE;TRACE 10;SLOPE RISE;  
COUNT 1;;MEASURE:CHANNEL10:  
DPROXIMAL:MODE PERCENT;  
PERCENT 10.0,50.0,90.0;  
UNIT -150.000E+00,0.0E+00,  
150.000E+00;;MEASURE:CHANNEL10:  
METHOD AUTO;;MEASURE:CHANNEL11:  
AMPLITUDE:STATE 0;;MEASURE:  
CHANNEL11:AVERAGE:STATE 0;;

```

MEASURE:CHANNEL11:AVGFREQ:STATE 0;;
MEASURE:CHANNEL11:AVGPERIOD:
STATE 0;;MEASURE:CHANNEL11:BWIDTH1:
STATE 0;;MEASURE:CHANNEL11:BWIDTH2:
STATE 0;;MEASURE:CHANNEL11:
DUTYCYCLE:STATE 0;;MEASURE:
CHANNEL11:FALL:STATE 0;;MEASURE:
CHANNEL11:FREQUENCY:STATE 0;;
MEASURE:CHANNEL11:HIG:STATE 0;;
MEASURE:CHANNEL11:LOW:STATE 0;;
MEASURE:CHANNEL11:MAXIMUM:STATE 0;;
MEASURE:CHANNEL11:MIDDLE:STATE 0;;
MEASURE:CHANNEL11:MINIMUM:STATE 0;;
MEASURE:CHANNEL11:NOVERSHOOT:
STATE 0;;MEASURE:CHANNEL11:NWIDTH:
STATE 0;;MEASURE:CHANNEL11:PERIOD:
STATE 0;;MEASURE:CHANNEL11:PNUMBER:
STATE 0;;MEASURE:CHANNEL11:
POVERSHOOT:STATE 0;;MEASURE:
CHANNEL11:PTOPEAK:STATE 0;;MEASURE:
CHANNEL11:PWID:STATE 0;;MEASURE:
CHANNEL11:RISE:STATE 0;;MEASURE:
CHANNEL11:RMS:STATE 0;;MEASURE:
CHANNEL11:SDEVIATION:STATE 0;;
MEASURE:CHANNEL11:TY1INTEG:
STATE 0;;MEASURE:CHANNEL11:
TY2INTEG:STATE 0;;MEASURE:
CHANNEL11:XY1INTEG:STATE 0;;
MEASURE:CHANNEL11:XY2INTEG:
STATE 0;;MEASURE:CHANNEL11:DELAY:
STATE OFF;MEASURE:SLOPE RISE;
COUNT 1;;MEASURE:CHANNEL11:DELAY:
REFERENCE:SOURCE TRACE;TRACE 11;
SLOPE RISE;COUNT 1;;MEASURE:
CHANNEL11:DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -150.000E+00,0.0E+00,
150.000E+00;;MEASURE:CHANNEL11:
METHOD AUTO;;MEASURE:CHANNEL12:
AMPLITUDE:STATE 0;;MEASURE:
CHANNEL12:AVERAGE:STATE 0;;MEASURE:
CHANNEL12:AVGFREQ:STATE 0;;MEASURE:
CHANNEL12:AVGPERIOD:STATE 0;;
MEASURE:CHANNEL12:BWIDTH1:STATE 0;;
MEASURE:CHANNEL12:BWIDTH2:STATE 0;;
MEASURE:CHANNEL12:DUTYCYCLE:
STATE 0;;MEASURE:CHANNEL12:FALL:
STATE 0;;MEASURE:CHANNEL12:
FREQUENCY:STATE 0;;MEASURE:
CHANNEL12:HIG:STATE 0;;MEASURE:
CHANNEL12:LOW:STATE 0;;MEASURE:
CHANNEL12:MAXIMUM:STATE 0;;MEASURE:
CHANNEL12:MIDDLE:STATE 0;;MEASURE:
CHANNEL12:MINIMUM:STATE 0;;MEASURE:
CHANNEL12:NOVERSHOOT:STATE 0;;
MEASURE:CHANNEL12:NWIDTH:STATE 0;;
MEASURE:CHANNEL12:PERIOD:STATE 0;;
MEASURE:CHANNEL12:PNUMBER:STATE 0;;
MEASURE:CHANNEL12:POVERSHOOT:
STATE 0;;MEASURE:CHANNEL12:PTOPEAK:
STATE 0;;MEASURE:CHANNEL12:PWID:
STATE 0;;MEASURE:CHANNEL12:RISE:
STATE 0;;MEASURE:CHANNEL12:RMS:
STATE 0;;MEASURE:CHANNEL12:
SDEVIATION:STATE 0;;MEASURE:
CHANNEL12:TY1INTEG:STATE 0;;
MEASURE:CHANNEL12:TY2INTEG:
STATE 0;;MEASURE:CHANNEL12:

```

```

XY1INTEG:STATE 0;;MEASURE:
CHANNEL12:XY2INTEG:STATE 0;;
MEASURE:CHANNEL12:DELAY:STATE OFF;
MEASURE:SLOPE RISE;COUNT 1;;
MEASURE:CHANNEL12:DELAY:
REFERENCE:SOURCE TRACE;TRACE 12;
SLOPE RISE;COUNT 1;;MEASURE:
CHANNEL12:DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -150.000E+00,0.0E+00,
150.000E+00;;MEASURE:CHANNEL12:
METHOD AUTO;;MEASURE:CHANNEL13:
AMPLITUDE:STATE 0;;MEASURE:
CHANNEL13:AVERAGE:STATE 0;;MEASURE:
CHANNEL13:AVGFREQ:STATE 0;;MEASURE:
CHANNEL13:AVGPERIOD:STATE 0;;
MEASURE:CHANNEL13:BWIDTH1:STATE 0;;
MEASURE:CHANNEL13:BWIDTH2:STATE 0;;
MEASURE:CHANNEL13:DUTYCYCLE:
STATE 0;;MEASURE:CHANNEL13:FALL:
STATE 0;;MEASURE:CHANNEL13:
FREQUENCY:STATE 0;;MEASURE:
CHANNEL13:HIG:STATE 0;;MEASURE:
CHANNEL13:LOW:STATE 0;;MEASURE:
CHANNEL13:MAXIMUM:STATE 0;;MEASURE:
CHANNEL13:MIDDLE:STATE 0;;MEASURE:
CHANNEL13:MINIMUM:STATE 0;;MEASURE:
CHANNEL13:NOVERSHOOT:STATE 0;;
MEASURE:CHANNEL13:NWIDTH:STATE 0;;
MEASURE:CHANNEL13:PERIOD:STATE 0;;
MEASURE:CHANNEL13:PNUMBER:STATE 0;;
MEASURE:CHANNEL13:POVERSHOOT:
STATE 0;;MEASURE:CHANNEL13:PTOPEAK:
STATE 0;;MEASURE:CHANNEL13:PWID:
STATE 0;;MEASURE:CHANNEL13:RISE:
STATE 0;;MEASURE:CHANNEL13:RMS:
STATE 0;;MEASURE:CHANNEL13:
SDEVIATION:STATE 0;;MEASURE:
CHANNEL13:TY1INTEG:STATE 0;;
MEASURE:CHANNEL13:TY2INTEG:
STATE 0;;MEASURE:CHANNEL13:
XY1INTEG:STATE 0;;MEASURE:
CHANNEL13:XY2INTEG:STATE
0;;MEASURE:CHANNEL13:DELAY:
STATE OFF;MEASURE:SLOPE RISE;
COUNT 1;;MEASURE:CHANNEL13:DELAY:
REFERENCE:SOURCE TRACE;TRACE 13;
SLOPE RISE;COUNT 1;;MEASURE:
CHANNEL13:DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -150.000E+00,0.0E+00,
150.000E+00;;MEASURE:CHANNEL13:
METHOD AUTO;;MEASURE:CHANNEL14:
AMPLITUDE:STATE
0;;MEASURE:CHANNEL14:AVERAGE:
STATE 0;;MEASURE:CHANNEL14:
AVGFREQ:STATE 0;;MEASURE:CHANNEL14:
AVGPERIOD:STATE 0;;MEASURE:
CHANNEL14:BWIDTH1:STATE 0;;MEASURE:
CHANNEL14:BWIDTH2:STATE 0;;MEASURE:
CHANNEL14:DUTYCYCLE:STATE 0;;
MEASURE:CHANNEL14:FALL:STATE 0;;
MEASURE:CHANNEL14:FREQUENCY:
STATE 0;;MEASURE:CHANNEL14:HIG:
STATE 0;;MEASURE:CHANNEL14:LOW:
STATE 0;;MEASURE:CHANNEL14:
MAXIMUM:STATE 0;;MEASURE:CHANNEL14:
MIDDLE:STATE 0;;MEASURE:CHANNEL14:

```

## 6.21 MEASure Group

MINIMUM:STATE 0;:MEASURE:CHANNEL14:  
NOVERSHOOT:STATE 0;:MEASURE:  
CHANNEL14:NWIDTH:STATE 0;:  
MEASURE:CHANNEL14:PERIOD:STATE 0;:  
MEASURE:CHANNEL14:PNUMBER:STATE 0;:  
MEASURE:CHANNEL14:POVERSHOOT:  
STATE 0;:MEASURE:CHANNEL14:PTOPEAK:  
STATE 0;:MEASURE:CHANNEL14:PWIDITH:  
STATE 0;:MEASURE:CHANNEL14:RISE:  
STATE 0;:MEASURE:CHANNEL14:RMS:  
STATE 0;:MEASURE:CHANNEL14:  
SDEVIATION:STATE 0;:MEASURE:  
CHANNEL14:TY1INTEG:STATE 0;:  
MEASURE:CHANNEL14:TY2INTEG:  
STATE 0;MEASURE:CHANNEL14:XY1INTEG:  
STATE 0;:MEASURE:CHANNEL14:  
XY2INTEG:STATE 0;:MEASURE:  
CHANNEL14:DELAY:STATE OFF;MEASURE:  
SLOPE RISE;COUNT 1;:MEASURE:  
CHANNEL14:DELAY:REFERENCE:  
SOURCE TRACE;TRACE 14;SLOPE RISE;  
COUNT 1;:MEASURE:CHANNEL14:  
DPROXIMAL:MODE PERCENT;  
PERCENT 10.0,50.0,90.0;  
UNIT -150.000E+00,0.0E+00,  
150.000E+00;:MEASURE:CHANNEL14:  
METHOD AUTO;:MEASURE:CHANNEL15:  
AMPLITUDE:STATE 0;:MEASURE:  
CHANNEL15:AVERAGE:STATE 0;:MEASURE:  
CHANNEL15:AVGFREQ:STATE 0;:MEASURE:  
CHANNEL15:AVGPERIOD:STATE 0;:  
MEASURE:CHANNEL15:BWIDITH1:STATE 0;:  
MEASURE:CHANNEL15:BWIDITH2:STATE 0;:  
MEASURE:CHANNEL15:DUTYCYCLE:  
STATE 0;:MEASURE:CHANNEL15:FALL:  
STATE 0;:MEASURE:CHANNEL15:  
FREQUENCY:STATE 0;:MEASURE:  
CHANNEL15:HIG:STATE 0;:MEASURE:  
CHANNEL15:LOW:STATE 0;:MEASURE:  
CHANNEL15:MAXIMUM:STATE 0;:MEASURE:  
CHANNEL15:MIDDLE:STATE 0;:MEASURE:  
CHANNEL15:MINIMUM:STATE 0;:MEASURE:  
CHANNEL15:NOVERSHOOT:STATE 0;:  
MEASURE:CHANNEL15:NWIDTH:STATE 0;:  
MEASURE:CHANNEL15:PERIOD:STATE 0;:  
MEASURE:CHANNEL15:PNUMBER:STATE 0;:  
MEASURE:CHANNEL15:POVERSHOOT:  
STATE 0;:MEASURE:CHANNEL15:PTOPEAK:  
STATE 0;:MEASURE:CHANNEL15:PWIDITH:  
STATE 0;:MEASURE:CHANNEL15:RISE:  
STATE 0;:MEASURE:CHANNEL15:RMS:  
STATE 0;:MEASURE:CHANNEL15:  
SDEVIATION:STATE 0;:MEASURE:  
CHANNEL15:TY1INTEG:STATE 0;:  
MEASURE:CHANNEL15:TY2INTEG:  
STATE 0;:MEASURE:CHANNEL15:  
XY1INTEG:STATE 0;:MEASURE:  
CHANNEL15:XY2INTEG:STATE 0;:  
MEASURE:CHANNEL15:DELAY:STATE OFF;  
MEASURE:SLOPE RISE;COUNT 1;:  
MEASURE:CHANNEL15:DELAY:REFERENCE:  
SOURCE TRACE;TRACE 15;SLOPE RISE;  
COUNT 1;:MEASURE:CHANNEL15:  
DPROXIMAL:MODE PERCENT;  
PERCENT 10.0,50.0,90.0;  
UNIT -15.000E+00,0.0E+00,  
15.000E+00;:MEASURE:CHANNEL15:  
METHOD AUTO;:MEASURE:CHANNEL16:

AMPLITUDE:STATE 0;:MEASURE:  
CHANNEL16:AVERAGE:STATE 0;:MEASURE:  
CHANNEL16:AVGFREQ:STATE 0;:MEASURE:  
CHANNEL16:AVGPERIOD:STATE 0;:  
MEASURE:CHANNEL16:BWIDITH1:STATE 0;:  
MEASURE:CHANNEL16:BWIDITH2:STATE 0;:  
MEASURE:CHANNEL16:DUTYCYCLE:  
STATE 0;:MEASURE:CHANNEL16:FALL:  
STATE 0;:MEASURE:CHANNEL16:  
FREQUENCY:STATE 0;:MEASURE:  
CHANNEL16:HIG:STATE 0;:MEASURE:  
CHANNEL16:LOW:STATE 0;:MEASURE:  
CHANNEL16:MAXIMUM:STATE 0;:MEASURE:  
CHANNEL16:MIDDLE:STATE 0;:MEASURE:  
CHANNEL16:MINIMUM:STATE 0;:MEASURE:  
CHANNEL16:NOVERSHOOT:STATE 0;:  
MEASURE:CHANNEL16:NWIDTH:STATE 0;:  
MEASURE:CHANNEL16:PERIOD:STATE 0;:  
MEASURE:CHANNEL16:PNUMBER:STATE 0;:  
MEASURE:CHANNEL16:POVERSHOOT:  
STATE 0;:MEASURE:CHANNEL16:PTOPEAK:  
STATE 0;:MEASURE:CHANNEL16:PWIDITH:  
STATE 0;:MEASURE:CHANNEL16:RISE:  
STATE 0;:MEASURE:CHANNEL16:RMS:  
STATE 0;:MEASURE:CHANNEL16:  
SDEVIATION:STATE 0;:MEASURE:  
CHANNEL16:TY1INTEG:STATE 0;:  
MEASURE:CHANNEL16:TY2INTEG:  
STATE 0;:MEASURE:CHANNEL16:  
XY1INTEG:STATE 0;:MEASURE:  
CHANNEL16:XY2INTEG:STATE 0;:  
MEASURE:CHANNEL16:DELAY:STATE OFF;  
MEASURE:SLOPE RISE;COUNT 1;:  
MEASURE:CHANNEL16:DELAY:REFERENCE:  
SOURCE TRACE;TRACE 16;SLOPE RISE;  
COUNT 1;:MEASURE:CHANNEL16:  
DPROXIMAL:MODE PERCENT;  
PERCENT 10.0,50.0,90.0;  
UNIT -15.000E+00,0.0E+00,  
15.000E+00;:MEASURE:CHANNEL16:  
METHOD AUTO;:MEASURE:MATH1:  
AMPLITUDE:STATE 0;:MEASURE:MATH1:  
AVERAGE:STATE 0;:MEASURE:MATH1:  
AVGFREQ:STATE 0;:MEASURE:MATH1:  
AVGPERIOD:STATE 0;:MEASURE:MATH1:  
BWIDITH1:STATE 0;:MEASURE:MATH1:  
BWIDITH2:STATE 0;:MEASURE:MATH1:  
DUTYCYCLE:STATE 0;:MEASURE:MATH1:  
FALL:STATE 0;:MEASURE:MATH1:  
FREQUENCY:STATE 0;:MEASURE:MATH1:  
HIGH:STATE 0;:MEASURE:MATH1:LOW:  
STATE 0;:MEASURE:MATH1:MAXIMUM:  
STATE 0;:MEASURE:MATH1:MIDDLE  
:STATE 0;:MEASURE:MATH1:MINIMUM:  
STATE 0;:MEASURE:MATH1:NOVERSHOOT:  
STATE 0;:MEASURE:MATH1:NWIDTH:  
STATE 0;:MEASURE:MATH1:PERIOD:  
STATE 0;:MEASURE:MATH1:PNUMBER:  
STATE 0;:MEASURE:MATH1:POVERSHOOT:  
STATE 0;:MEASURE:MATH1:PTOPEAK:  
STATE 0;:MEASURE:MATH1:PWIDITH:  
STATE 0;:MEASURE:MATH1:RISE:  
STATE 0;:MEASURE:MATH1:RMS:  
STATE 0;:MEASURE:MATH1:SDEVIATION:  
STATE 0;:MEASURE:MATH1:TY1INTEG:  
STATE 0;:MEASURE:MATH1:TY2INTEG:  
STATE 0;:MEASURE:MATH1:XY1INTEG:  
STATE 0;:MEASURE:MATH1:XY2INTEG:

```

STATE 0 ; :MEASURE:MATH1:DELAY:
STATE OFF;MEASURE:SLOPE RISE;
COUNT 1 ; :MEASURE:MATH1:DELAY:
REFERENCE:SOURCE TRACE;TRACE MATH1;
SLOPE RISE;COUNT 1 ; :MEASURE:MATH1:
DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -3.00,0.00,3.00 ; :MEASURE:
MATH1:METHOD AUTO ; :MEASURE:MATH2:
AMPLITUDE:STATE 0 ; :MEASURE:MATH2:
AVERAGE:STATE 0 ; :MEASURE:MATH2:
AVGFREQ:STATE 0 ; :MEASURE:MATH2:
AVGPERIOD:STATE 0 ; :MEASURE:MATH2:
BWIDTH1:STATE 0 ; :MEASURE:MATH2:
BWIDTH2:STATE 0 ; :MEASURE:MATH2:
DUTYCYCLE:STATE 0 ; :MEASURE:MATH2:
FALL:STATE 0 ; :MEASURE:MATH2:
FREQUENCY:STATE 0 ; :MEASURE:MATH2:
HIGH:STATE 0 ; :MEASURE:MATH2:LOW:
STATE 0 ; :MEASURE:MATH2:MAXIMUM:
STATE 0 ; :MEASURE:MATH2:MIDDLE:
STATE 0 ; :MEASURE:MATH2:MINIMUM:
STATE 0 ; :MEASURE:MATH2:NOVERSHOOT:
STATE 0 ; :MEASURE:MATH2:NWIDTH:
STATE 0 ; :MEASURE:MATH2:PERIOD:
STATE 0 ; :MEASURE:MATH2:PNUMBER:
STATE 0 ; :MEASURE:MATH2:POVERSHOOT:
STATE 0 ; :MEASURE:MATH2:PTOPEAK:
STATE 0 ; :MEASURE:MATH2:PWIDITH:
STATE 0 ; :MEASURE:MATH2:RISE:
STATE 0 ; :MEASURE:MATH2:RMS:
STATE 0 ; :MEASURE:MATH2:SDEVIATION:
STATE 0 ; :MEASURE:MATH2:TY1INTEG:
STATE 0 ; :MEASURE:MATH2:TY2INTEG:
STATE 0 ; :MEASURE:MATH2:XY1INTEG:
STATE 0 ; :MEASURE:MATH2:XY2INTEG:
STATE 0 ; :MEASURE:MATH2:DELAY:
STATE OFF;MEASURE:SLOPE RISE;
COUNT 1 ; :MEASURE:MATH2:DELAY:
REFERENCE:SOURCE TRACE;TRACE MATH2;
SLOPE RISE;COUNT 1 ; :MEASURE:MATH2:
DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -3.00,0.00,3.00 ; :MEASURE:
MATH2:METHOD AUTO ; :MEASURE:MATH3:
AMPLITUDE:STATE 0 ; :MEASURE:MATH3:
AVERAGE:STATE 0 ; :MEASURE:MATH3:
AVGFREQ:STATE 0 ; :MEASURE:MATH3:
AVGPERIOD:STATE 0 ; :MEASURE:MATH3:
BWIDTH1:STATE 0 ; :MEASURE:MATH3:
BWIDTH2:STATE 0 ; :MEASURE:MATH3:
DUTYCYCLE:STATE 0 ; :MEASURE:MATH3:
FALL:STATE 0 ; :MEASURE:MATH3:
FREQUENCY:STATE 0 ; :MEASURE:MATH3:
HIGH:STATE 0 ; :MEASURE:MATH3:LOW:
STATE 0 ; :MEASURE:MATH3:MAXIMUM:
STATE 0 ; :MEASURE:MATH3:MIDDLE:
STATE 0 ; :MEASURE:MATH3:MINIMUM:
STATE 0 ; :MEASURE:MATH3:NOVERSHOOT:
STATE 0 ; :MEASURE:MATH3:NWIDTH:
STATE 0 ; :MEASURE:MATH3:PERIOD:
STATE 0 ; :MEASURE:MATH3:PNUMBER:
STATE 0 ; :MEASURE:MATH3:POVERSHOOT:
STATE 0 ; :MEASURE:MATH3:PTOPEAK:
STATE 0 ; :MEASURE:MATH3:PWIDITH:
STATE 0 ; :MEASURE:MATH3:RISE:
STATE 0 ; :MEASURE:MATH3:RMS:
STATE 0 ; :MEASURE:MATH3:SDEVIATION:

```

```

STATE 0 ; :MEASURE:MATH3:TY1INTEG:
STATE 0 ; :MEASURE:MATH3:TY2INTEG:
STATE 0 ; :MEASURE:MATH3:XY1INTEG:
STATE 0 ; :MEASURE:MATH3:XY2INTEG:
STATE 0 ; :MEASURE:MATH3:DELAY:
STATE OFF;MEASURE:SLOPE RISE;
COUNT 1 ; :MEASURE:MATH3:DELAY:
REFERENCE:SOURCE TRACE;TRACE MATH2;
SLOPE RISE;COUNT 1 ; :MEASURE:MATH3:
DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -3.00,0.00,3.00 ; :MEASURE:
MATH3:METHOD AUTO ; :MEASURE:MATH4:
AMPLITUDE:STATE 0 ; :MEASURE:MATH4:
AVERAGE:STATE 0 ; :MEASURE:MATH4:
AVGFREQ:STATE 0 ; :MEASURE:MATH4:
AVGPERIOD:STATE 0 ; :MEASURE:MATH4:
BWIDTH1:STATE 0 ; :MEASURE:MATH4:
BWIDTH2:STATE 0 ; :MEASURE:MATH4:
DUTYCYCLE:STATE 0 ; :MEASURE:MATH4:
FALL:STATE 0 ; :MEASURE:MATH4:
FREQUENCY:STATE 0 ; :MEASURE:MATH4:
HIGH:STATE 0 ; :MEASURE:MATH4:LOW:
STATE 0 ; :MEASURE:MATH4:MAXIMUM:
STATE 0 ; :MEASURE:MATH4:MIDDLE:
STATE 0 ; :MEASURE:MATH4:MINIMUM:
STATE 0 ; :MEASURE:MATH4:NOVERSHOOT:
STATE 0 ; :MEASURE:MATH4:NWIDTH:
STATE 0 ; :MEASURE:MATH4:PERIOD:
STATE 0 ; :MEASURE:MATH4:PNUMBER:
STATE 0 ; :MEASURE:MATH4:POVERSHOOT:
STATE 0 ; :MEASURE:MATH4:PTOPEAK:
STATE 0 ; :MEASURE:MATH4:PWIDITH:
STATE 0 ; :MEASURE:MATH4:RISE:
STATE 0 ; :MEASURE:MATH4:RMS:
STATE 0 ; :MEASURE:MATH4:SDEVIATION:
STATE 0 ; :MEASURE:MATH4:TY1INTEG:
STATE 0 ; :MEASURE:MATH4:TY2INTEG:
STATE 0 ; :MEASURE:MATH4:XY1INTEG:
STATE 0 ; :MEASURE:MATH4:XY2INTEG:
STATE 0 ; :MEASURE:MATH4:DELAY:
STATE OFF;MEASURE:SLOPE RISE;
COUNT 1 ; :MEASURE:MATH4:DELAY:
REFERENCE:SOURCE TRACE;TRACE MATH2;
SLOPE RISE;COUNT 1 ; :MEASURE:MATH4:
DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -3.00,0.00,3.00 ; :MEASURE:
MATH4:METHOD AUTO ; :MEASURE:MATH5:
AMPLITUDE:STATE 0 ; :MEASURE:MATH5:
AVERAGE:STATE 0 ; :MEASURE:MATH5:
AVGFREQ:STATE 0 ; :MEASURE:MATH5:
AVGPERIOD:STATE 0 ; :MEASURE:MATH5:
BWIDTH1:STATE 0 ; :MEASURE:MATH5:
BWIDTH2:STATE 0 ; :MEASURE:MATH5:
DUTYCYCLE:STATE 0 ; :MEASURE:MATH5:
FALL:STATE 0 ; :MEASURE:MATH5:
FREQUENCY:STATE 0 ; :MEASURE:MATH5:
HIGH:STATE 0 ; :MEASURE:MATH5:LOW:
STATE 0 ; :MEASURE:MATH5:MAXIMUM:
STATE 0 ; :MEASURE:MATH5:MIDDLE:
STATE 0 ; :MEASURE:MATH5:MINIMUM:
STATE 0 ; :MEASURE:MATH5:NOVERSHOOT:
STATE 0 ; :MEASURE:MATH5:NWIDTH:
STATE 0 ; :MEASURE:MATH5:PERIOD:
STATE 0 ; :MEASURE:MATH5:PNUMBER:
STATE 0 ; :MEASURE:MATH5:POVERSHOOT:
STATE 0 ; :MEASURE:MATH5:PTOPEAK:

```

## 6.21 MEASure Group

```
STATE 0;:MEASURE:MATH5:PWIDTh:
STATE 0;:MEASURE:MATH5:RISE:
STATE 0;:MEASURE:MATH5:RMS:
STATE 0;:MEASURE:MATH5:SDEVIATION:
STATE 0;:MEASURE:MATH5:TY1INTEG:
STATE 0;:MEASURE:MATH5:TY2INTEG:
STATE 0;:MEASURE:MATH5:XY1INTEG:
STATE 0;:MEASURE:MATH5:XY2INTEG:
STATE 0;:MEASURE:MATH5:DELAY:
STATE OFF;MEASURE:SLOPE RISE;
COUNT 1;:MEASURE:MATH5:DELAY:
REFERENCE:SOURCE TRACE;TRACE MATH2;
SLOPE RISE;COUNT 1;:MEASURE:MATH5:
DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -3.00,0.00,3.00;:MEASURE:
MATH5:METHOD AUTO;:MEASURE:MATH6:
AMPLITUDE:STATE 0;:MEASURE:MATH6:
AVERAGE:STATE 0;:MEASURE:MATH6:
AVGFREQ:STATE 0;:MEASURE:MATH6:
AVGPERIOD:STATE 0;:MEASURE:MATH6:
BWIDTh1:STATE 0;:MEASURE:MATH6:
BWIDTh2:STATE 0;:MEASURE:MATH6:
DUTYCYCLE:STATE 0;:MEASURE:MATH6:
FALL:STATE 0;:MEASURE:MATH6:
FREQUENCY:STATE 0;:MEASURE:MATH6:
HIGH:STATE 0;:MEASURE:MATH6:LOW:
STATE 0;:MEASURE:MATH6:MAXIMUM:
STATE 0;:MEASURE:MATH6:MIDDLE:
STATE 0;:MEASURE:MATH6:MINIMUM:
STATE 0;:MEASURE:MATH6:NOVERSHOOT:
STATE 0;:MEASURE:MATH6:NWIDTh:
STATE 0;:MEASURE:MATH6:PERIOD:
STATE 0;:MEASURE:MATH6:PNUMBER:
STATE 0;:MEASURE:MATH6:POVERSHOOT:
STATE 0;:MEASURE:MATH6:PTOPEAK:
STATE 0;:MEASURE:MATH6:PWIDTh:
STATE 0;:MEASURE:MATH6:RISE:
STATE 0;:MEASURE:MATH6:RMS:
STATE 0;:MEASURE:MATH6:SDEVIATION:
STATE 0;:MEASURE:MATH6:TY1INTEG:
STATE 0;:MEASURE:MATH6:TY2INTEG:
STATE 0;:MEASURE:MATH6:XY1INTEG:
STATE 0;:MEASURE:MATH6:XY2INTEG:
STATE 0;:MEASURE:MATH6:DELAY:
STATE OFF;MEASURE:SLOPE RISE;
COUNT 1;:MEASURE:MATH6:DELAY:
REFERENCE:SOURCE TRACE;TRACE MATH2;
SLOPE RISE;COUNT 1;:MEASURE:MATH6:
DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -3.00,0.00,3.00;:MEASURE:
MATH6:METHOD AUTO;:MEASURE:MATH7:
AMPLITUDE:STATE 0;:MEASURE:MATH7:
AVERAGE:STATE 0;:MEASURE:MATH7:
AVGFREQ:STATE 0;:MEASURE:MATH7:
AVGPERIOD:STATE 0;:MEASURE:MATH7:
BWIDTh1:STATE 0;:MEASURE:MATH7:
BWIDTh2:STATE 0;:MEASURE:MATH7:
DUTYCYCLE:STATE 0;:MEASURE:MATH7:
FALL:STATE 0;:MEASURE:MATH7:
FREQUENCY:STATE 0;:MEASURE:MATH7:
HIGH:STATE 0;:MEASURE:MATH7:LOW:
STATE 0;:MEASURE:MATH7:MAXIMUM:
STATE 0;:MEASURE:MATH7:MIDDLE:
STATE 0;:MEASURE:MATH7:MINIMUM:
STATE 0;:MEASURE:MATH7:NOVERSHOOT:
STATE 0;:MEASURE:MATH7:NWIDTh:
```

```
STATE 0;:MEASURE:MATH7:PERIOD:
STATE 0;:MEASURE:MATH7:PNUMBER:
STATE 0;:MEASURE:MATH7:POVERSHOOT:
STATE 0;:MEASURE:MATH7:PTOPEAK:
STATE 0;:MEASURE:MATH7:PWIDTh:
STATE 0;:MEASURE:MATH7:RISE:
STATE 0;:MEASURE:MATH7:RMS:
STATE 0;:MEASURE:MATH7:SDEVIATION:
STATE 0;:MEASURE:MATH7:TY1INTEG:
STATE 0;:MEASURE:MATH7:TY2INTEG:
STATE 0;:MEASURE:MATH7:XY1INTEG:
STATE 0;:MEASURE:MATH7:XY2INTEG:
STATE 0;:MEASURE:MATH7:DELAY:
STATE OFF;MEASURE:SLOPE RISE;
COUNT 1;:MEASURE:MATH7:DELAY:
REFERENCE:SOURCE TRACE;TRACE MATH2;
SLOPE RISE;COUNT 1;:MEASURE:MATH7:
DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -3.00,0.00,3.00;:MEASURE:
MATH7:METHOD AUTO;:MEASURE:MATH8:
AMPLITUDE:STATE 0;:MEASURE:MATH8:
AVERAGE:STATE 0;:MEASURE:MATH8:
AVGFREQ:STATE 0;:MEASURE:MATH8:
AVGPERIOD:STATE 0;:MEASURE:MATH8:
BWIDTh1:STATE 0;:MEASURE:MATH8:
BWIDTh2:STATE 0;:MEASURE:MATH8:
DUTYCYCLE:STATE 0;:MEASURE:MATH8:
FALL:STATE 0;:MEASURE:MATH8:
FREQUENCY:STATE 0;:MEASURE:MATH8:
HIGH:STATE 0;:MEASURE:MATH8:LOW:
STATE 0;:MEASURE:MATH8:MAXIMUM:
STATE 0;:MEASURE:MATH8:MIDDLE:
STATE 0;:MEASURE:MATH8:MINIMUM:
STATE 0;:MEASURE:MATH8:NOVERSHOOT:
STATE 0;:MEASURE:MATH8:NWIDTh:
STATE 0;:MEASURE:MATH8:PERIOD:
STATE 0;:MEASURE:MATH8:PNUMBER:
STATE 0;:MEASURE:MATH8:POVERSHOOT:
STATE 0;:MEASURE:MATH8:PTOPEAK:
STATE 0;:MEASURE:MATH8:PWIDTh:
STATE 0;:MEASURE:MATH8:RISE:
STATE 0;:MEASURE:MATH8:RMS:
STATE 0;:MEASURE:MATH8:SDEVIATION:
STATE 0;:MEASURE:MATH8:TY1INTEG:
STATE 0;:MEASURE:MATH8:TY2INTEG:
STATE 0;:MEASURE:MATH8:XY1INTEG:
STATE 0;:MEASURE:MATH8:XY2INTEG:
STATE 0;:MEASURE:MATH8:DELAY:
STATE OFF;MEASURE:SLOPE RISE;
COUNT 1;:MEASURE:MATH8:DELAY:
REFERENCE:SOURCE TRACE;TRACE MATH2;
SLOPE RISE;COUNT 1;:MEASURE:MATH8:
DPROXIMAL:MODE PERCENT;
PERCENT 10.0,50.0,90.0;
UNIT -3.00,0.00,3.00;:MEASURE:
MATH8:METHOD AUTO;:MEASURE:
CYCLEMODE 0;
TRANGE -5.0000000,5.0000000
```

**:MEASure: {CHANnel<x> | MATH<x>}?**

**Function** Turns ON/OFF all the parameters of the waveform or queries the current setting.

**Syntax** :MEASure: {CHANnel<x> | MATH<x>}?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8

**Example** (The following is an example for CH1.)  
:MEASURE: CHANNEL1? -> :MEASURE:  
CHANNEL1:AMPLITUDE:STATE 0;;  
MEASURE:CHANNEL1:AVERAGE:STATE 0;;  
MEASURE:CHANNEL1:AVGFREQ:STATE 0;;  
MEASURE:CHANNEL1:AVGPERIOD:  
STATE 0;;MEASURE:CHANNEL1:BWIDTH1:  
STATE 0;;MEASURE:CHANNEL1:BWIDTH2:  
STATE 0;;MEASURE:CHANNEL1:  
DUTYCYCLE:STATE 0;;MEASURE:  
CHANNEL1:FALL:STATE 0;;MEASURE:  
CHANNEL1:FREQUENCY:STATE 0;;  
MEASURE:CHANNEL1:HIG:STATE 0;;  
MEASURE:CHANNEL1:LOW:STATE 0;;  
MEASURE:CHANNEL1:MAXIMUM:STATE 0;;  
MEASURE:CHANNEL1:MIDDLE:STATE 0;;  
MEASURE:CHANNEL1:MINIMUM:STATE 0;;  
MEASURE:CHANNEL1:NOVERSHOOT:  
STATE 0;;MEASURE:CHANNEL1:NWIDTH:  
STATE 0;;MEASURE:CHANNEL1:PERIOD:  
STATE 0;;MEASURE:CHANNEL1:PNUMBER:  
STATE 0;;MEASURE:CHANNEL1:  
POVERSHOOT:STATE 0;;MEASURE:  
CHANNEL1:PTOPEAK:STATE 0;;MEASURE:  
CHANNEL1:PWIDTH:STATE 0;;MEASURE:  
CHANNEL1:RISE:STATE 0;;MEASURE:  
CHANNEL1:RMS:STATE 0;;MEASURE:  
CHANNEL1:SDEVIATION:STATE 0;;  
MEASURE:CHANNEL1:TY1INTEG:STATE 0;;  
MEASURE:CHANNEL1:TY2INTEG:STATE 0;;  
MEASURE:CHANNEL1:XY1INTEG:STATE 0;;  
MEASURE:CHANNEL1:XY2INTEG:STATE 0;;  
MEASURE:CHANNEL1:DELAY:STATE OFF;  
MEASURE:SLOPE RISE;COUNT 1;;  
MEASURE:CHANNEL1:DELAY:  
REFERENCE:SOURCE TRACE;TRACE 1;  
SLOPE RISE;COUNT 1;;MEASURE:  
CHANNEL1:DPROXIMAL:MODE PERCENT;  
PERCENT 10.0,50.0,90.0;  
UNIT -150.000E+00,0.0E+00,  
150.000E+00;;MEASURE:CHANNEL1:  
METHOD AUTO

**:MEASure: {CHANnel<x> | MATH<x>} :ALL**

**Function** Turns ON/OFF collectively all the measurement items of the channel.

**Syntax** :MEASure: {CHANnel<x> | MATH<x>} :ALL  
{<Boolean>}  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8

**Example** (The following is an example for CH1.)  
:MEASURE: CHANNEL1:ALL OFF

**:MEASure: {CHANnel<x> | MATH<x>} :COPY**

**Function** Copies the ON/OFF setting of all measurement items of a specified channel to all other channels.

**Syntax** :MEASure: {CHANnel<x> | MATH<x>} :COPY  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8

**Example** (The following is an example of copying the ON/OFF setting of all items of CH1 to all other channels.)  
:MEASURE: CHANNEL1: COPY

**:MEASure: {CHANnel<x> | MATH<x>} :DELAy?**

**Function** Queries all settings of the source waveform of the delay between channels.

**Syntax** :MEASure: {CHANnel<x> | MATH<x>} :  
DELAy?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8

**Example** :MEASURE: CHANNEL1:DELAy? ->  
:MEASURE: CHANNEL1:DELAy: STATE OFF;  
MEASURE:SLOPE RISE;COUNT 1;  
MEASURE: CHANNEL1:DELAy: REFERENCE:  
SOURCE TRACE;TRACE 1;SLOPE RISE;  
COUNT 1

**:MEASure: {CHANnel<x> | MATH<x>} :DELAy:MEASure?**

**Function** Queries all settings of the source waveform of the delay between channels.

**Syntax** :MEASure: {CHANnel<x> | MATH<x>} :  
DELAy:MEASure?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8

**Example** :MEASURE: CHANNEL1:DELAy:MEASure? ->  
:MEASURE: CHANNEL1:DELAy:MEASure:  
SLOPE RISE;COUNT 1

**:MEASure: {CHANnel<x> | MATH<x>} :DELAy:MEASure:COUNT**

**Function** Sets the number of counts of the source waveform of the delay between channels.

**Syntax** :MEASure: {CHANnel<x> | MATH<x>} :  
DELAy:MEASure:COUNT {<NRf>}  
:MEASure: {CHANnel<x> | MATH<x>} :  
DELAy:MEASure:COUNT?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
<NRf>=1 to 9

**Example** :MEASURE: CHANNEL1:DELAy:MEASure:  
COUNT 2  
:MEASURE: CHANNEL1:DELAy:MEASure:  
COUNT? -> :MEASURE: CHANNEL1:DELAy:  
MEASure:COUNT 2

## 6.21 MEASure Group

**:MEASure: {CHANnel<x> | MATH<x>}**  
**:DElay:MEASure:SLOPe**  
Function Sets the slope of the source waveform of the delay between channels.  
Syntax :MEASure: {CHANnel<x> | MATH<x>}:  
DElay:MEASure:SLOPe {RISE|FALL}  
:MEASure: {CHANnel<x> | MATH<x>}:  
DElay:MEASure:SLOPe?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
Example :MEASURE:CHANNEL1:DELAY:MEASURE:  
SLOPE RISE  
:MEASURE:CHANNEL1:DELAY:MEASURE:  
SLOPE? -> :MEASURE:CHANNEL1:DELAY:  
MEASURE:SLOPE RISE

**:MEASure: {CHANnel<x> | MATH<x>}**  
**:DElay:REFerence?**  
Function Queries all settings of the reference waveform of the delay between channels.  
Syntax :MEASure: {CHANnel<x> | MATH<x>}:  
DElay:REFerence?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
Example :MEASURE:CHANNEL1:DELAY:REFERENCE?  
-> :MEASURE:CHANNEL1:DELAY:  
REFERENCE:SOURCE TRACE;TRACE 1;  
SLOPE RISE;COUNT 1

**:MEASure: {CHANnel<x> | MATH<x>}**  
**:DElay:REFerence:COUNT**  
Function Sets the number of counts of the reference waveform of the delay between channels.  
Syntax :MEASure: {CHANnel<x> | MATH<x>}:  
DElay:REFerence:COUNT {<Nrf>}  
:MEASure: {CHANnel<x> | MATH<x>}:  
DElay:REFerence:COUNT?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
<Nrf>=1 to 9  
Example :MEASURE:CHANNEL1:DELAY:REFERENCE:  
COUNT 2  
:MEASURE:CHANNEL1:DELAY:REFERENCE:  
COUNT? -> :MEASURE:CHANNEL1:DELAY:  
REFERENCE:COUNT 2

**:MEASure: {CHANnel<x> | MATH<x>}**  
**:DElay:REFerence:SLOPe**  
Function Sets the slope of the reference waveform of the delay between channels.  
Syntax :MEASure: {CHANnel<x> | MATH<x>}:  
DElay:REFerence:SLOPe {RISE|FALL}  
:MEASure: {CHANnel<x> | MATH<x>}:  
DElay:REFerence:SLOPe?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
Example :MEASURE:CHANNEL1:DELAY:REFERENCE:  
SLOPE RISE  
:MEASURE:CHANNEL1:DELAY:REFERENCE:  
SLOPE? -> :MEASURE:CHANNEL1:DELAY:  
REFERENCE:SLOPE RISE

**:MEASure: {CHANnel<x> | MATH<x>}**  
**:DElay:REFerence:SOURce**  
Function Sets the reference of the delay between channels to the trigger point or the waveform or queries the current setting.  
Syntax :MEASure: {CHANnel<x> | MATH<x>}:  
DElay:REFerence:SOURce  
{TRACe|TRIGGer}  
:MEASure: {CHANnel<x> | MATH<x>}:  
DElay:REFerence:SOURce?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
Example :MEASURE:CHANNEL1:DELAY:REFERENCE:  
SOURCE TRACE  
:MEASURE:CHANNEL1:DELAY:REFERENCE:  
SOURCE? -> :MEASURE:CHANNEL1:DELAY:  
REFERENCE:SOURCE TRACE

**:MEASure: {CHANnel<x> | MATH<x>}**  
**:DElay:REFerence:TRACe**  
Function Sets the trace of the reference waveform of the delay between channels.  
Syntax :MEASure: {CHANnel<x> | MATH<x>}:  
DElay:REFerence:TRACe  
{<Nrf> | MATH<x>}  
:MEASure: {CHANnel<x> | MATH<x>}:  
DElay:REFerence:TRACe?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
<Nrf>=1 to 16  
Example :MEASURE:CHANNEL1:DELAY:REFERENCE:  
TRACE 2  
:MEASURE:CHANNEL1:DELAY:REFERENCE:  
TRACE? -> :MEASURE:CHANNEL1:DELAY:  
REFERENCE:TRACE 2

**:MEASure: {CHANnel<x> | MATH<x>}**  
**:DElay:STATe**  
Function Sets the display format of the delay parameter or queries the current setting.  
Syntax :MEASure: {CHANnel<x> | MATH<x>}:  
DElay:STATe {TIME|OFF|DEGRee}  
:MEASure: {CHANnel<x> | MATH<x>}:  
DElay:STATe?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
Example :MEASURE:CHANNEL1:DELAY:STATE TIME  
:MEASURE:CHANNEL1:DELAY:STATE? ->  
:MEASURE:CHANNEL1:DELAY:STATE TIME

**:MEASure: {CHANnel<x> | MATH<x>}:DElay:VALue?**  
Function Queries the value of the delay parameter of the waveform that is measured automatically.  
Syntax :MEASure: {CHANnel<x> | MATH<x>}:  
DElay:VALue?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
Example :MEASURE:CHANNEL1:DELAY:VALUE? ->  
:MEASURE:CHANNEL1:DELAY:  
VALUE -24.420000E-03  
Description If the measurement is not possible, "NAN (Not A Number)" is returned.

**:MEASure:{CHANnel<x>|MATH<x>}  
:DPRoximal?**

**Function** Queries all settings related to distal, mesial, and proximal.

**Syntax** :MEASure:{CHANnel<x>|MATH<x>}:  
DPRoximal?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8

**Example** :MEASURE:CHANNEL1:DPROXIMAL? ->  
:MEASURE:CHANNEL1:DPROXIMAL:  
MODE PERCENT;  
PERCENT 10.0,50.0,90.0;  
UNIT -150.000E+00,0.0E+00,  
150.000E+00

**:MEASure:{CHANnel<x>|MATH<x>}  
:DPRoximal:MODE**

**Function** Sets the mode of the distal, mesial, and proximal points or queries the current setting.

**Syntax** :MEASure:{CHANnel<x>|MATH<x>}:  
DPRoximal:MODE {PERCent|UNIT}  
:MEASure:{CHANnel<x>|MATH<x>}:  
DPRoximal:MODE?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8

**Example** :MEASURE:CHANNEL1:DPROXIMAL:  
MODE PERCENT  
:MEASURE:CHANNEL1:DPROXIMAL:MODE?  
-> :MEASURE:CHANNEL1:DPROXIMAL:  
MODE PERCENT

**:MEASure:{CHANnel<x>|MATH<x>}  
:DPRoximal:PERCent**

**Function** Sets the distal, mesial, and proximal points in percentage or queries the current setting.

**Syntax** :MEASure:{CHANnel<x>|MATH<x>}:  
DPRoximal:PERCent  
{<NRf>,<NRf>,<NRf>}  
:MEASure:{CHANnel<x>|MATH<x>}:  
DPRoximal:PERCent?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
<NRf>=0 to 100 (% , 1 steps)  
(In the order <proximal>,<mesial>,<distal>)

**Example** :MEASURE:CHANNEL1:DPROXIMAL:  
PERCENT 40,60,80  
:MEASURE:CHANNEL1:DPROXIMAL:  
PERCENT? -> :MEASURE:CHANNEL1:  
DPROXIMAL:PERCENT 40.0,60.0,80.0

**:MEASure:{CHANnel<x>|MATH<x>}  
:DPRoximal:UNIT**

**Function** Sets the distal, mesial, and proximal points or queries the current setting.

**Syntax** :MEASure:{CHANnel<x>|MATH<x>}:  
DPRoximal:UNIT  
{<Voltage>,<Voltage>,<Voltage>|  
<Current>,<Current>,<Current>|  
<NRf>,<NRf>,<NRf>}  
:MEASure:{CHANnel<x>|MATH<x>}:  
DPRoximal:UNIT?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
For CHANnel<x>: Voltage, <Current>,  
<NRf>=The selectable range varies  
depending on the range and offset settings.  
For details, see the *DL750 User's Manual*.  
For MATH<x>: <NRf>=ls±5 div (0.01 div steps)  
(In the order <proximal>,<mesial>,<distal>)

**Example** :MEASURE:CHANNEL1:DPROXIMAL:  
UNIT -50V,0V,50V  
:MEASURE:CHANNEL1:DPROXIMAL:UNIT?  
-> :MEASURE:CHANNEL1:DPROXIMAL:  
UNIT -50.000E+00,0.0E+00,50.000E+00

**Description** When the target is MATH<x>, only <NRf> can be specified for the unit.

**:MEASure:{CHANnel<x>|MATH<x>}:METHod**

**Function** Sets the High and Low points or queries the current setting.

**Syntax** :MEASure:{CHANnel<x>|MATH<x>}:  
METHod {AUTO|MAXMin}  
:MEASure:{CHANnel<x>|MATH<x>}:  
METHod?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8

**Example** :MEASURE:CHANNEL1:METHod AUTO  
:MEASURE:CHANNEL1:METHod? ->  
:MEASURE:CHANNEL1:METHod AUTO

**:MEASure:{CHANnel<x>|MATH<x>}  
:<Parameter>?**

**Function** Queries all settings related to the parameter of the waveform.

**Syntax** :MEASure:{CHANnel<x>|MATH<x>}:  
<Parameter>?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
<Parameter>={AMPLitude|AVERage|  
AVGFreq|AVGPeriod|BWIDth1|BWIDth2|  
DUTYcycle|FALL|FREQuency|HIGH|LOW|  
MAXimum|MIDDLE|MINimum|NOVershoot|  
NWIDth|PERiod|PNUMber|POVershoot|  
PTOPeak|PWIDth|RISE|RMS|SDEVIation|  
TY1Integ|TY2Integ|XY1Integ|XY2Integ}

**Example** (The following is an example for the average value of CH1.)  
:MEASURE:CHANNEL1:AVERAGE? ->  
:MEASURE:CHANNEL1:AVERAGE:STATE 0



## 6.21 MEASure Group

### **:MEASure:{CHANnel<x>|MATH<x>}**

#### **:<Parameter>:STATE**

**Function** Turns ON/OFF the parameter of the waveform one by one or queries the current setting.

**Syntax** :MEASure:{CHANnel<x>|MATH<x>}:<Parameter>:STATE {<Boolean>}  
:MEASure:{CHANnel<x>|MATH<x>}:<Parameter>:STATE?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
<Parameter>={AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth1|BWIDth2|DUTYcycle|FALL|FREQuency|HIGH|LOW|MAXimum|MIDDLE|MINimum|NOVershoot|NWIDth|PERiod|PNUMBER|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEVIation|TY1Integ|TY2Integ|XY1Integ|XY2Integ}

**Example** (The following is an example for the average value of CH1.)  
:MEASURE:CHANNEL1:AVERAGE:STATE ON  
:MEASURE:CHANNEL1:AVERAGE:STATE? ->  
:MEASURE:CHANNEL1:AVERAGE:STATE 1

### **:MEASure:{CHANnel<x>|MATH<x>}**

#### **:<Parameter>:VALUE?**

**Function** Queries the value of the parameter of the waveform that is measured automatically.

**Syntax** :MEASure:{CHANnel<x>|MATH<x>}:<Parameter>:VALUE?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8  
<Parameter>={AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth1|BWIDth2|DUTYcycle|FALL|FREQuency|HIGH|LOW|MAXimum|MIDDLE|MINimum|NOVershoot|NWIDth|PERiod|PNUMBER|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEVIation|TY1Integ|TY2Integ|XY1Integ|XY2Integ}

**Example** (The following is an example for the average value of CH1.)  
:MEASURE:CHANNEL1:AVERAGE:VALUE? ->  
:MEASURE:CHANNEL1:AVERAGE:  
VALUE 115.95507E-03

**Description** If the measurement is not possible, "NAN" (Not A Number) is returned.

### **:MEASure:CYCLEmode**

**Function** Sets the 1 cycle mode of the automated measurement of waveform parameters or queries the current setting.

**Syntax** :MEASure:CYCLEmode {<Boolean>}  
:MEASure:CYCLEmode?

**Example** :MEASURE:CYCLEMODE ON  
:MEASURE:CYCLEMODE? -> :MEASURE:  
CYCLEMODE 1

### **:MEASure:MODE**

**Function** Turns ON/OFF the automated measurement or queries the current setting.

**Syntax** :MEASure:MODE {<Boolean>}  
:MEASure:MODE?

**Example** :MEASURE:MODE ON  
:MEASURE:MODE? -> :MEASURE:MODE 1

### **:MEASure:TRANge (Time Range)**

**Function** Sets the measurement range or queries the current setting.

**Syntax** :MEASure:TRANge {<NRF>,<NRF>}  
:MEASure:TRANge?  
<NRF>=-5 to 5 divisions (10 divisions/displayed record length steps)

**Example** :MEASURE:TRANGE -4,4  
:MEASURE:TRANGE? -> :MEASURE:  
TRANGE -4.0000000,4.0000000

### **:MEASure:WAIT?**

**Function** Waits for the completion of the automated measurement with timeout.

**Syntax** :MEASure:WAIT? {<NRF>}  
<NRF>= 1 to 360000 (timeout time, in units of 100 ms)

**Example** :MEASURE:WAIT? 100 -> :MEASURE:  
WAIT 1

**Description**

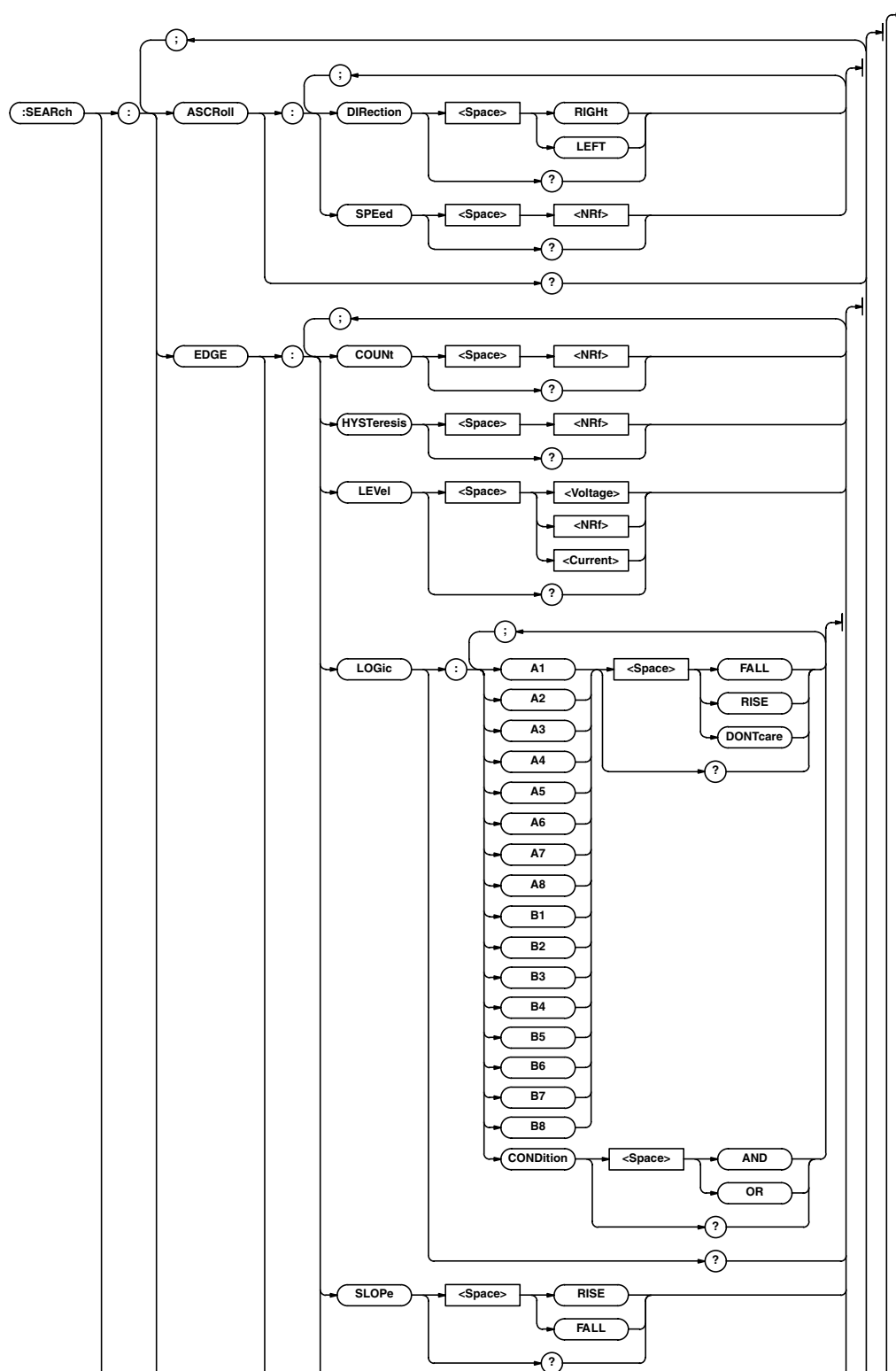
- If the execution of the automated measurement completes within the timeout time, 0 is returned; if it is not complete or automated measurement is not being executed, 1 is returned.
- Even if the timeout time is set long, 0 is returned when the execution of the automated measurement is complete.

## 6.22 SEARCh Group

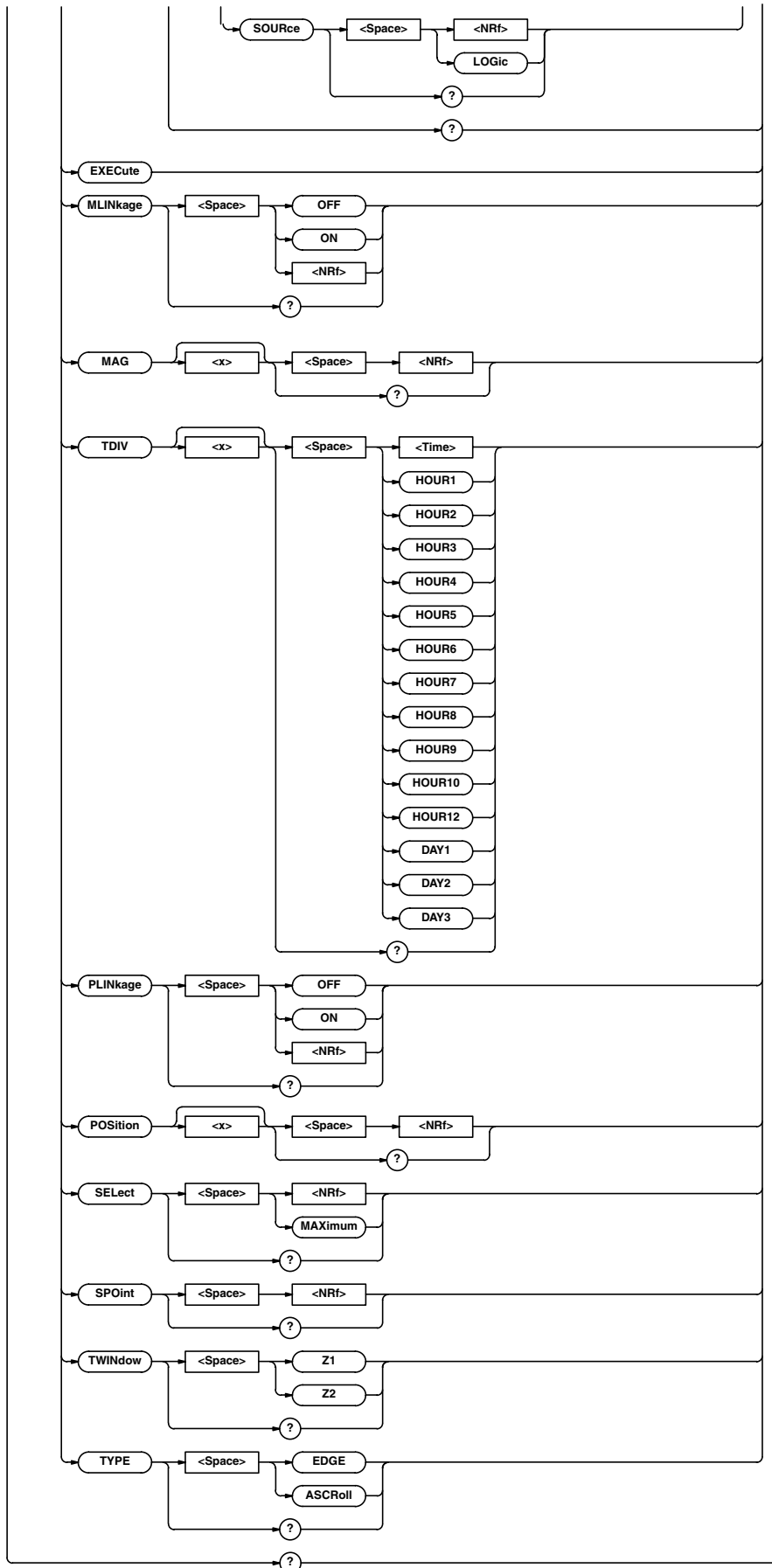
The commands in this group are used to search and extract an arbitrary pattern section from the acquired waveform. You can make the same settings, execute the same operations, and make the same inquiries as when SHIFT+ZOOM on the front panel is used.

When the time axis range setting is as follows, the selectable range varies depending on the record length setting. For details, see "Selectable Range of Cursor Position" in section 8.1 of the DL750 User's Manual.

<NRf>=-5 to 5 div



6.22 SEARCh Group



**:SEARCH?**

Function Queries all settings related to the search function.

Syntax :SEARCH?

Example :SEARCH? -> :SEARCH:TYPE EDGE;EDGE:SOURCE 1;LEVEL 0.0E+00;SLOPE RISE;HYSTERESIS MIDDLE;COUNT 1;:SEARCH:ASCROLL:DIRECTION RIGHT;SPEED 4;:SEARCH:TWINDOW Z1;MLINKAGE 1;MAG1 500.0E-06;MAG2 500.0E-06;PLINKAGE 0;POSITION1 0.0000000;POSITION2 0.0000000;SPOINT -5.0000000;SELECT NAN

**:SEARCH:ASCROLL?**

Function Queries all settings related to the auto scroll.

Syntax :SEARCH:ASCROLL?

Example :SEARCH:ASCROLL? -> :SEARCH:ASCROLL:DIRECTION RIGHT;SPEED 10

**:SEARCH:ASCROLL:DIRECTION**

Function Sets the direction for moving the zoom box of the auto scroll or queries the current setting.

Syntax :SEARCH:ASCROLL:DIRECTION {LEFT|RIGHT}

Example :SEARCH:ASCROLL:DIRECTION LEFT

**:SEARCH:ASCROLL:SPEED**

Function Sets the speed for moving the zoom box of the auto scroll or queries the current setting.

Syntax :SEARCH:ASCROLL:SPEED {<Nrf>}  
:SEARCH:ASCROLL:SPEED?  
<Nrf>=1 to 10

Example :SEARCH:ASCROLL:SPEED 1

**:SEARCH:EDGE?**

Function Queries all settings related to the edge search.

Syntax :SEARCH:EDGE?

Example :SEARCH:EDGE? -> :SEARCH:EDGE:SOURCE 1;LEVEL 0.0E+00;SLOPE RISE;HYSTERESIS MIDDLE;COUNT 1

**:SEARCH:EDGE:COUNT**

Function Sets the edge search count or queries the current setting.

Syntax :SEARCH:EDGE:COUNT {<Nrf>}  
:SEARCH:EDGE:COUNT?  
<Nrf>=1 to 1000000

Example :SEARCH:EDGE:COUNT 100

**:SEARCH:EDGE:HYSTERESIS**

Function Sets the hysteresis of the edge search level or queries the current setting.

Syntax :SEARCH:EDGE:HYSTERESIS {HIGH|LOW|MIDDLE}

Example :SEARCH:EDGE:HYSTERESIS MIDDLE

**:SEARCH:EDGE:LEVEL**

Function Sets the edge search level or queries the current setting.

Syntax :SEARCH:EDGE:LEVEL {<Voltage>|<Current>|<Nrf>}  
:SEARCH:EDGE:LEVEL?  
{<Voltage>|<Current>|<Nrf>}=The selectable range varies depending on the range and offset settings. For details, see the *DL750 User's Manual*.

Example :SEARCH:EDGE:LEVEL -1V

Description Use voltage or current when :SEARCH:EDGE:SOURCE is set to a channel; use <Nrf> when :SEARCH:EDGE:SOURCE is set to MATH.

**:SEARCH:EDGE:LOGIC?**

Function Queries all settings related to the logic channel of the edge search.

Syntax :SEARCH:EDGE:LOGIC?

Example :SEARCH:EDGE:LOGIC? -> :SEARCH:EDGE:LOGIC:A1 RISE;A2 DONTCARE;A3 DONTCARE;A4 DONTCARE;A5 DONTCARE;A6 DONTCARE;A7 DONTCARE;A8 DONTCARE;B1 DONTCARE;B2 DONTCARE;B3 DONTCARE;B4 DONTCARE;B5 DONTCARE;B6 DONTCARE;B7 DONTCARE;B8 DONTCARE;CONDITION OR

Description This command is valid when :SEARCH:EDGE:SOURCE is set to LOGIC.

**:SEARCH:EDGE:LOGIC:{A1|A2|A3|A4|A5|A6|A7|A8|B1|B2|B3|B4|B5|B6|B7|B8}**

Function Sets the slope of the bit of the logic channel of the edge search or queries the current setting.

Syntax :SEARCH:EDGE:LOGIC:{A1|A2|A3|A4|A5|A6|A7|A8|B1|B2|B3|B4|B5|B6|B7|B8} {FALL|RISE|DONTcare}

Example :SEARCH:EDGE:LOGIC:A1 RISE

Description This command is valid when :SEARCH:EDGE:SOURCE is set to LOGIC.

**:SEARCH:EDGE:LOGIC:CONDITION**

Function Sets the condition of the bit of the logic channel of the edge search or queries the current setting.

Syntax :SEARCH:EDGE:LOGIC:CONDITION {AND|OR}

Example :SEARCH:EDGE:LOGIC:CONDITION AND

Description This command is valid when :SEARCH:EDGE:SOURCE is set to LOGIC.

## 6.22 SEARCh Group

### **:SEARCh:EDGE:SLOPe**

Function Sets the edge polarity or queries the current setting.

Syntax :SEARCh:EDGE:SLOPe {FALL|RISE}  
:SEARCh:EDGE:SLOPe?

Example :SEARCH:EDGE:SLOPE RISE  
:SEARCH:EDGE:SLOPE? -> :SEARCH:  
EDGE:SLOPE RISE

### **:SEARCh:EDGE:SOURce**

Function Sets the source waveform of the edge search or queries the current setting.

Syntax :SEARCh:EDGE:SOURce {<NRf>|LOGic}  
:SEARCh:EDGE:SOURce?  
<NRf>=1 to 16

Example :SEARCH:EDGE:SOURCE 1  
:SEARCH:EDGE:SOURCE? -> :SEARCH:  
EDGE:SOURCE 1

### **:SEARCh:EXECute**

Function Executes the search. This is an overlap command.

Syntax :SEARCh:EXECute

Example :SEARCH:EXECUTE

### **:SEARCh:MAG<x>**

Function Sets the zoom rate of the zoom window or queries the current setting.

Syntax :SEARCh:TDIV<x> {<NRf>}  
:SEARCh:TDIV<x>?  
<x>=1 or 2  
<NRf>=2 to 100000000 (see the *DL750 User's Manual*)

Example :SEARCH:MAG1 10  
:SEARCH:MAG1? -> :SEARCH:MAG1 10.0

Description When zooming a waveform measured using the external clock, set the magnification.

### **:SEARCh:MLINKage (Mag Linkage)**

Function Turns ON/OFF the link to the T/div value of the zoom waveform or queries the current setting.

Syntax :SEARCh:MLINKage {<Boolean>}  
:SEARCh:MLINKage?

Example :SEARCH:MLINKAGE ON  
:SEARCH:MLINKAGE? -> :SEARCH:  
MLINKAGE 0

### **:SEARCh:PLINKage (Position Linkage)**

Function Turns ON/OFF the link when moving the zone box or queries the current setting.

Syntax :SEARCh:PLINKage {<Boolean>}  
:SEARCh:PLINKage?

Example :SEARCH:PLINKAGE OFF  
:SEARCH:PLINKAGE? -> :SEARCH:  
PLINKAGE 0

### **:SEARCh:POSition<x>**

Function Sets the zoom box position or queries the current setting.

Syntax :SEARCh:POSition<x> {<NRf>}  
:SEARCh:POSition<x>?  
<x>=1 or 2  
<NRf>=-5 to 5 div

Example :SEARCH:POSITION1 0  
:SEARCH:POSITION1? -> :SEARCH:  
POSITION1 0.0000000

### **:SEARCh:SElect**

Function Sets the storage number of the searched position or queries the position stored at the storage number.

Syntax :SEARCh:SElect {<NRf>|MAXimum}  
:SEARCh:SElect?  
<NRf>=1 to 1000

Example :SEARCH:SELECT 1  
:SEARCH:SELECT? -> :SEARCH:  
SELECT 5.00997453

Description If there is no searched position, "NAN" is returned for the query.

### **:SEARCh:SElect? MAXimum**

Function Queries the maximum storage number of searched positions.

Syntax :SEARCh:SElect? {MAXimum}

Example :SEARCH:SELECT? MAXIMUM -> :SEARCH:  
SELECT 1

Description If there is no searched position, "NAN" is returned for the query.

### **:SEARCh:SPOint (Start Point)**

Function Sets the search start point or queries the current setting.

Syntax :SEARCh:SPOint {<NRf>}  
:SEARCh:SPOint?  
<NRf>=-5 to 5 div

Example :SEARCH:SPOINT 3  
:SEARCH:SPOINT? -> :SEARCH:  
SPOINT 3.0000000

### **:SEARCh:TDIV<x>**

Function Sets the T/div value of the zoom waveform or queries the current setting.

Syntax :SEARCh:TDIV<x>  
{<Time>|HOUR1|HOUR2|HOUR3|HOUR4|HOUR5|  
HOUR6|HOUR7|HOUR8|HOUR9|HOUR10|  
HOUR12|DAY1|DAY2|DAY3}  
:SEARCh:TDIV<x>?  
<x>=1 or 2  
<Time>=See the *DL750 User's Manual*

Example :SEARCH:TDIV1 200ms  
:SEARCH:TDIV1? -> :SEARCH:  
TDIV1 200.0E-03

Description When zooming in on the waveforms measured using the internal clock, set the magnification using a T/div value.

### **:SEARCh:TWINDow (Target WINDOW)**

Function Sets the zoom window on which the searched section is to be displayed or queries the current setting.

Syntax :SEARCh:TWINDow {Z1|Z2}  
:SEARCh:TWINDow?

Example :SEARCH:TWINDOW Z1  
:SEARCH:TWINDOW? -> :SEARCH:  
TWINDOW Z1

### **:SEARCh:TYPE**

Function Sets the search type or queries the current setting.

Syntax :SEARCh:TYPE {EDGE|ASCROLL}

Example :SEARCH:TYPE EDGE  
:SEARCH:TYPE? -> :SEARCH:TYPE EDGE

## 6.23 SNAP Group

The commands in this group are used to execute the snapshot. You can execute the same operation as the SNAP SHOT key on the front panel.

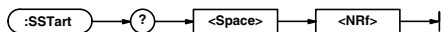


### :SNAP

Function Executes the snapshot.  
 Syntax :SNAP  
 Example :SNAP  
 Description Clear trace is carried out using CLear.

## 6.24 SStart Group

The commands in this group executes single start. Sets the trigger mode to Single and starts the waveform acquisition.



### :SStart (Single Start)

Function Executes single start.  
 Syntax :SStart  
 Example :SSTART

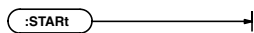
### :SStart?

Function Executes single start and waits for the completion.  
 Syntax :SStart? {<NRf>}  
 <NRf>=1 to 360000 (100 ms unit: wait time, START and wait)  
 0 (START only. No wait.)  
 -36000 to -1 (100 ms unit: wait time, do not START and wait)  
 Example :SSTART? 100 -> :SSTART 0  
 Description

- If the specified time period is positive, data acquisition is started in the SINGLE TRIGGER mode and waits for the operation to stop.
- If the specified time period is 0, data acquisition is started and 0 is returned without waiting for the operation to stop.
- If the specified time period is negative (-), the instrument simply waits for the operation to stop. Data acquisition is not started.

## 6.25 START Group

The command in this group is used to start the waveform acquisition. You can execute the same operation as the START/STOP key on the front panel.



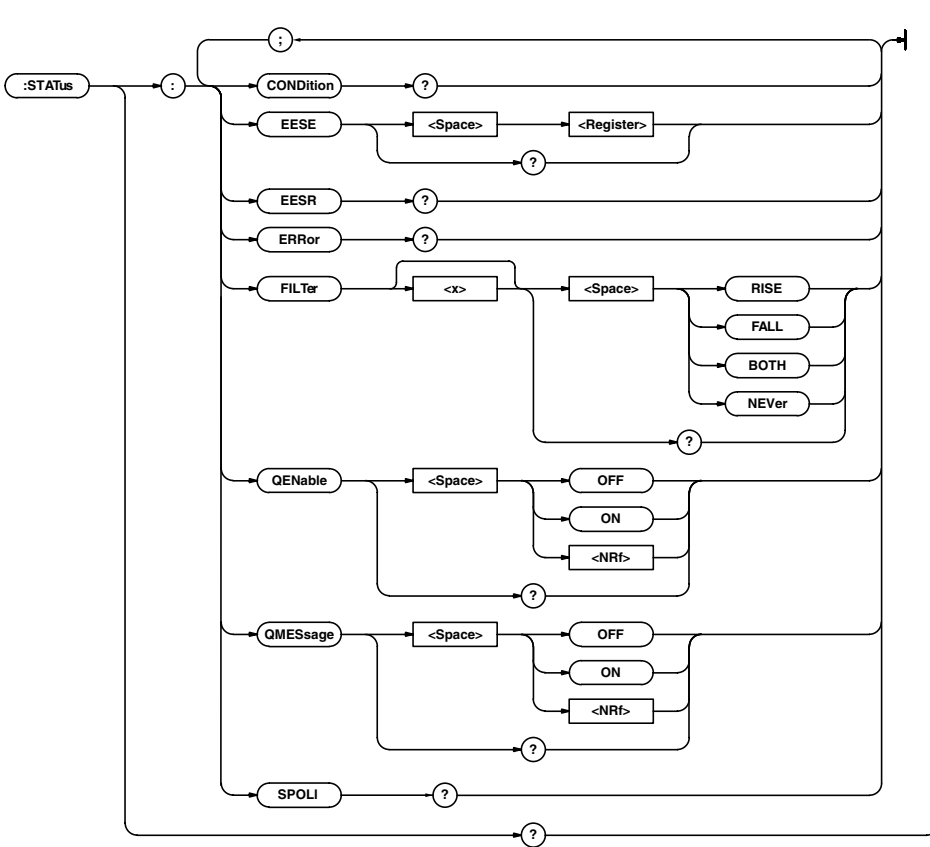
### :START

Function Starts the waveform acquisition.  
 Syntax :START  
 Example :START  
 Description

- Use STOP to stop the waveform acquisition.
- To perform realtime recording, use the COMMunicate:OVERlap command to apply the overlap command for bit 6 (during file access) before starting/stopping.

## 6.26 STATus Group

The commands in this group are used to make settings and inquiries related to the status report. There are no front panel keys that correspond to the commands in this group. For details on the status report, see chapter 7.



### **:STATus?**

**Function** Queries all settings related to the communication status 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;  
 QENABLE 0;QMESSAGE 1

### **:STATus:CONDition?**

**Function** Queries the contents of the condition register.

**Syntax** :STATus:CONDition?

**Example** :STATus:CONDition? -> 16

**Description** For the description regarding how to synchronize the program using :STATus:CONDition, see page 5-8.

### **:STATus:EESE (Extended Event Status Enable register)**

**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 #B00000000  
 :STATus:EESE? -> :STATus:EESE 0

### **:STATus:EESR? (Extended Event Status Register)**

**Function** Queries the content of the extended event register and clears the register.

**Syntax** :STATus:EESR?

**Example** :STATus:EESR? -> 0

### **:STATus:ERRor?**

**Function** Queries the error code and message information (top of the error queue).

**Syntax** :STATus:ERRor?

**Example** :STATus:ERRor? -> 901, "Backup failure"

**Description**

- When there is no error, 0 (No error) is returned.
- The message cannot be returned in Japanese.
- You can specify whether or not to add the message using the "STATus:QMESSage" command.

**:STATUS:FILTer<x>**

Function	Sets the 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	Specify how each bit of the condition register is to change to set the event. If RISE is specified, the event is set when the bit changes from 0 to 1.

**:STATUS:QENable**

Function	Sets whether or not to store messages other than errors to the error queue (ON/OFF) or queries the current setting.
Syntax	:STATUS:QENable {<Boolean>} :STATUS:QENable?
Example	:STATUS:QENABLE ON :STATUS:QENABLE? -> :STATUS: QENABLE 1

**:STATUS:QMESsage**

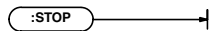
Function	Sets whether or not to attach message information to the response to the STATUS:Error? query (ON/OFF) or queries the current setting.
Syntax	:STATUS:QMESsage {<Boolean>} :STATUS:QMESsage?
Example	:STATUS:QMESSAGE ON :STATUS:QMESSAGE? -> :STATUS: QMESSAGE 1

**:STATUS:SPOLI? (Serial Poll)**

Function	Executes serial polling.
Syntax	:STATUS:SPOLI?
Example	:STATUS:SPOLL? -> :STATUS:SPOLL 0
Description	This command is dedicated to the RS-232 interface. For GP-IB, this function is provided as an interface message.

**6.27 STOP Group**

The command in this group is used to start the waveform acquisition. You can execute the same operation as the START/STOP key on the front panel.

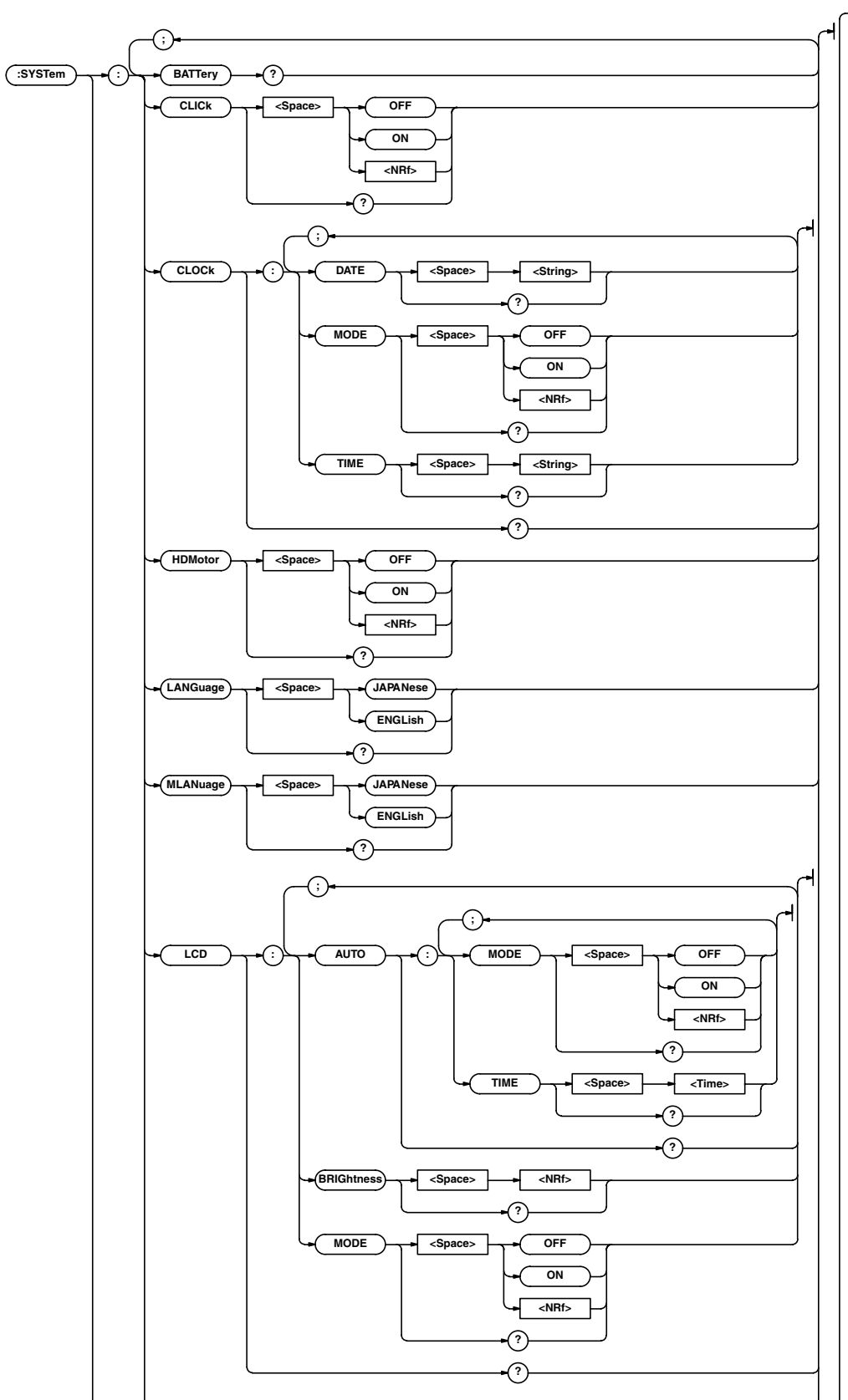
**:STOP**

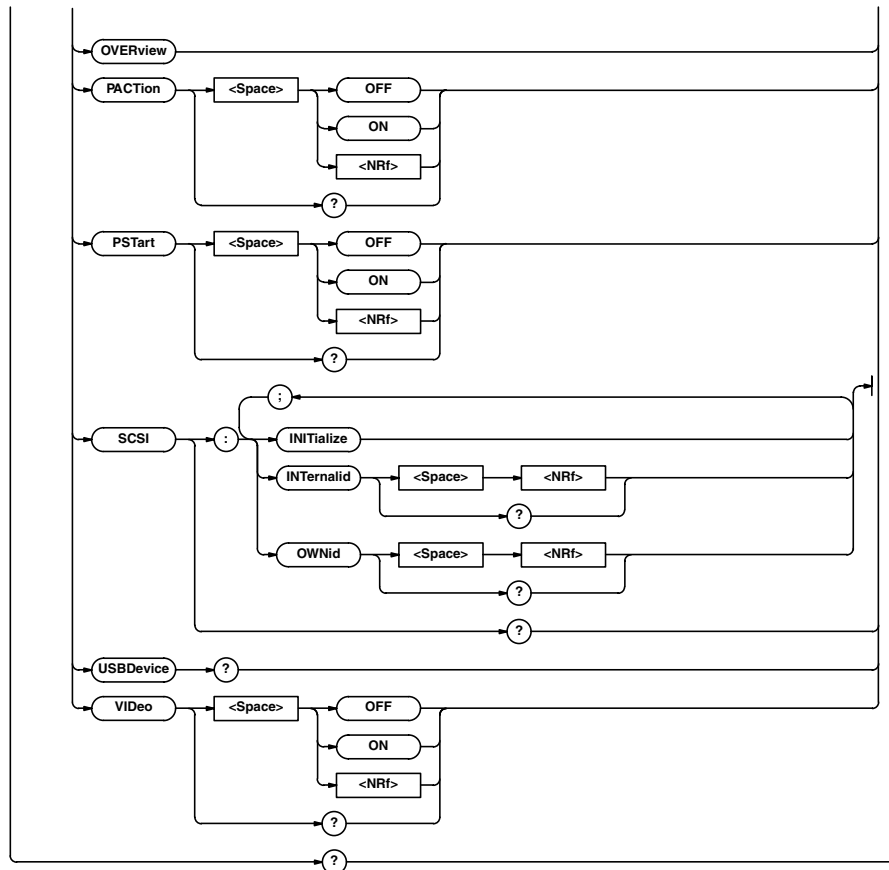
Function	Stops the waveform acquisition.
Syntax	:STOP
Example	:STOP
Description	<ul style="list-style-type: none"> <li>Use START to start the waveform acquisition.</li> <li>To perform realtime recording, use the COMMunicate:OVERlap command to apply the overlap command for bit 6 (during file access) before starting/stopping.</li> </ul>



### 6.28 SYSTEM Group

The commands in this group deal with the system. You can make the same settings and inquiries as when the System Config menu of MISC on the front panel is used.



**: SYSTEM?**

Function Queries all settings related to the system.  
 Syntax :SYSTEM?  
 Example :SYSTEM? -> :SYSTEM:SCSI:  
 INTERNALID 4;OWNID 6;:SYSTEM:CLOCK:  
 MODE 1;DATE "2002/06/07";  
 TIME "15:01:24";:SYSTEM:  
 LANGUAGE JAPANESE;CLICK 1;  
 HDMOTOR 1;MLANUAGE ENGLISH;  
 PACTION 0;PSTART 0;VIDEO 1;LCD:  
 MODE 1;BRIGHTNESS 4;AUTO:MODE 0;  
 TIME 1;:SYSTEM:OCANCEL 0

**: SYSTEM:CLICK**

Function Turns ON/OFF the click sound or queries the current setting.  
 Syntax :SYSTEM:CLICK {<Boolean>}  
 :SYSTEM:CLICK?  
 Example :SYSTEM:CLICK ON  
 :SYSTEM:CLICK? -> :SYSTEM:CLICK 1

**: SYSTEM:CLOCK?**

Function Queries all settings related to the date and time.  
 Syntax :SYSTEM:CLOCK?  
 Example :SYSTEM:CLOCK? -> :SYSTEM:CLOCK:  
 MODE 1;DATE "2002/06/06";  
 TIME "16:37:50"

**: SYSTEM:CLOCK:DATE**

Function Sets the date or queries the current setting.  
 Syntax :SYSTEM:CLOCK:DATE <String>  
 :SYSTEM:CLOCK:DATE?  
 <String>=YYYY/MM/DD, see the *DL750 User's Manual*  
 Example :SYSTEM:CLOCK:DATE "2002/06/04"  
 :SYSTEM:CLOCK:DATE? -> :SYSTEM:  
 CLOCK:DATE "2002/06/04"

**: SYSTEM:CLOCK:MODE**

Function Turns ON/OFF the date/time display 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

**: SYSTEM:CLOCK:TIME**

Function Sets the time or queries the current setting.  
 Syntax :SYSTEM:CLOCK:TIME <String>  
 :SYSTEM:CLOCK:TIME?  
 <String>=HH:MM:SS, see the *DL750 User's Manual*  
 Example :SYSTEM:CLOCK:TIME "14:30:00"  
 :SYSTEM:CLOCK:TIME? -> :SYSTEM:  
 CLOCK:TIME "14:30:00"

## 6.28 SYSTem Group

### **:SYSTem:HDMotor**

Function Turns ON/OFF the motor of the internal hard disk or queries the current setting.

Syntax :SYSTem:HDMotor {<Boolean>}  
:SYSTem:HDMotor?

Example :SYSTEM:HDMOTOR ON  
:SYSTEM:HDMOTOR? -> :SYSTEM:  
HDMOTOR 1

Description An error occurs if the internal hard disk (optional) is not installed.

### **:SYSTem:LANGUage**

Function Sets the message language or queries the current setting.

Syntax :SYSTem:LANGUage {ENGLish|JAPANese}  
:SYSTem:LANGUage?

Example :SYSTEM:LANGUAGE JAPANESE  
:SYSTEM:LANGUAGE? -> :SYSTEM:  
LANGUAGE JAPANESE

### **:SYSTem:LCD?**

Function Queries all settings related to the LCD.

Syntax :SYSTem:LCD?

Example :SYSTEM:LCD? -> :SYSTEM:LCD:MODE 1;  
BRIGHTNESS 2;AUTO:MODE 0;TIME 1

### **:SYSTem:LCD:AUTO?**

Function Queries all settings related to the auto off function of the LCD backlight.

Syntax :SYSTem:LCD:AUTO?

Example :SYSTEM:LCD:AUTO? -> :SYSTEM:LCD:  
AUTO:MODE 0;TIME 1

### **:SYSTem:LCD:AUTO:MODE**

Function Turns ON/OFF the auto off function of the LCD backlight or queries the current setting.

Syntax :SYSTem:LCD:AUTO:MODE {<Boolean>}  
:SYSTem:LCD:AUTO:MODE?

Example :SYSTEM:LCD:AUTO:MODE ON  
:SYSTEM:LCD:AUTO:MODE? -> :SYSTEM:  
LCD:AUTO:MODE 1

### **:SYSTem:LCD:AUTO:TIME**

Function Sets the auto off time of the LCD backlight or queries the current setting.

Syntax :SYSTem:LCD:AUTO:TIME {<NRf>}  
:SYSTem:LCD:AUTO:TIME?  
<NRf>=1 to 60 (minutes)

Example :SYSTEM:LCD:AUTO:TIME 1  
:SYSTEM:LCD:AUTO:TIME? -> :SYSTEM:  
LCD:AUTO:TIME 1

### **:SYSTem:LCD:BRIGhtness**

Function Sets the brightness of the LCD or queries the current setting.

Syntax :SYSTem:LCD:BRIGhtness {<NRf>}  
:SYSTem:LCD:BRIGhtness?  
<NRf>=0 to 5

Example :SYSTEM:LCD:BRIGHTNESS 2  
:SYSTEM:LCD:BRIGHTNESS? -> :SYSTEM:  
LCD:BRIGHTNESS 2

### **:SYSTem:LCD:MODE**

Function Turns ON/OFF the LCD backlight 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

### **:SYSTem:MLAnGuage**

Function Sets the menu language or queries the current setting.

Syntax :SYSTem:MLAnGuage  
{ENGLish|JAPANese}  
:SYSTem:MLAnGuage?

Example :SYSTEM:MLANGUAGE JAPANESE  
:SYSTEM:MLANGUAGE? -> :SYSTEM:  
MLANGUAGE JAPANESE

### **:SYSTem:OVERview**

Function Displays system information.

Syntax :SYSTem:OVERview

Example :SYSTEM:OVERVIEW

### **:SYSTem:PACTion**

Function Sets whether to enable the action-on-trigger mode at power ON or queries the current setting.

Syntax :SYSTem:PACTion {<Boolean>}  
:SYSTem:PACTion?

Example :SYSTEM:PACTION ON  
:SYSTEM:PACTION? -> :SYSTEM:  
PACTION 1

### **:SYSTem:PStart**

Function Sets whether to start the waveform acquisition at power ON or queries the current setting.

Syntax :SYSTem:PStart {<Boolean>}  
:SYSTem:PStart?

Example :SYSTEM:PSTART ON  
:SYSTEM:PSTART? -> :SYSTEM:PSTART 1

### **:SYSTem:SCSI?**

Function Queries all settings related to the SCSI-ID.

Syntax :SYSTem:SCSI?

Example :SYSTEM:SCSI? -> :SYSTEM:SCSI:  
INTERNALID 4;OWNID 6

### **:SYSTem:SCSI:INITialize**

Function Executes the initialization of SCSI related parameters.

Syntax :SYSTem:SCSI:INITialize

Example :SYSTEM:SCSI:INITIALIZE

Description If you changed the SCSI-ID of the DL750 using the ":SYSTem:SCSI:OWNid" command, make sure to issue this command.

### **:SYSTem:SCSI:INTernalid**

Function Set the SCSI-ID of the internal hard disk or queries the current settings.

Syntax :SYSTem:SCSI:INTernalid {<NRf>}  
:SYSTem:SCSI:INTernalid?  
<NRf>=0 to 7

Example :SYSTEM:SCSI:INTERNALID 4  
:SYSTEM:SCSI:INTERNALID? ->  
:SYSTEM:SCSI:INTERNALID 4

**:SYSTEM:SCSI:OWNid**

Function Set the SCSI-ID of the DL750 or queries the current settings.

Syntax :SYSTEM:SCSI:OWNid {<NRf>}  
:SYSTEM:SCSI:OWNid?  
<NRf>=0 to 7

Example :SYSTEM:SCSI:OWNID 7  
:SYSTEM:SCSI:OWNID? -> :SYSTEM:  
SCSI:OWNID 7

**:SYSTEM:VIDeo**

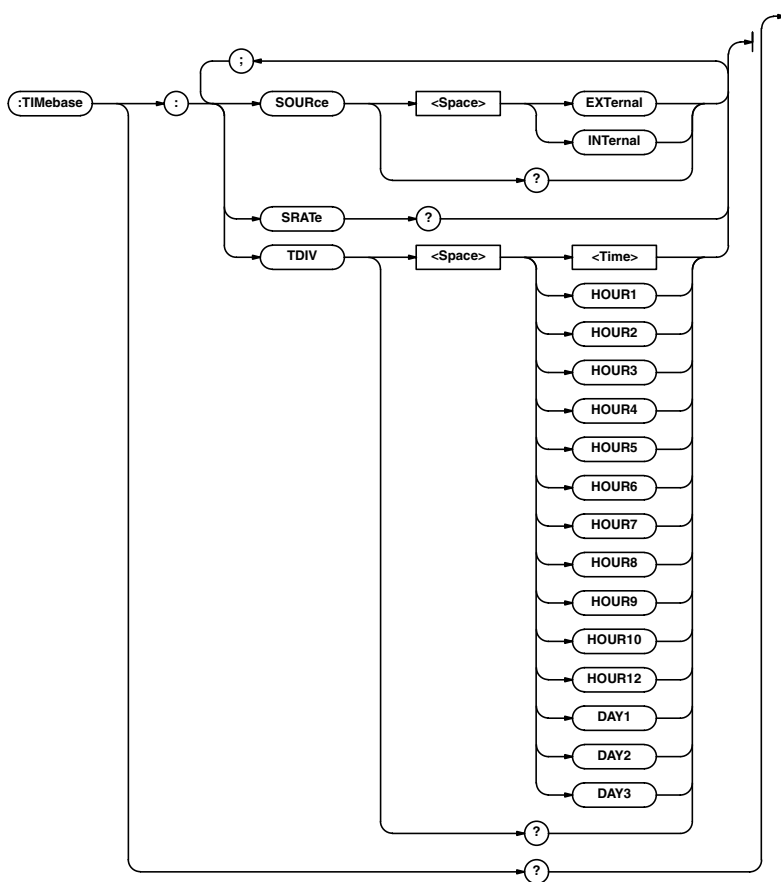
Function Turns ON/OFF the video output or queries the current setting.

Syntax :SYSTEM:VIDeo {<Boolean>}  
:SYSTEM:VIDeo?

Example :SYSTEM:VIDEO ON  
:SYSTEM:VIDEO? -> :SYSTEM:VIDEO 1

## 6.29 TIMEbase Group

The commands in this group deal with the time base. You can make the same settings and inquiries as when the TIME/DIV knob on the front panel is used.



### **:TIMEbase?**

Function Queries all settings related to the time base.

Syntax :TIMEbase?

Example :TIMEBASE? -> :TIMEBASE:  
SOURCE INTERNAL;TDIV 2.000E+00

### **:TIMEbase:SOURCE**

Function Sets the time base or queries the current setting.

Syntax :TIMEbase:SOURCE  
{EXTERNAL | INTERNAL}  
:TIMEbase:SOURCE?

Example :TIMEBASE:SOURCE INTERNAL  
:TIMEBASE:SOURCE? -> :TIMEBASE:  
SOURCE INTERNAL

### **:TIMEbase:SRATE? (Sample RATE)**

Function Queries the sample rate or queries the current setting.

Syntax :TIMEbase:SRATE?

Example :TIMEBASE:SRATE? -> :TIMEBASE:  
SRATE 10.0E+06

### **:TIMEbase:TDIV**

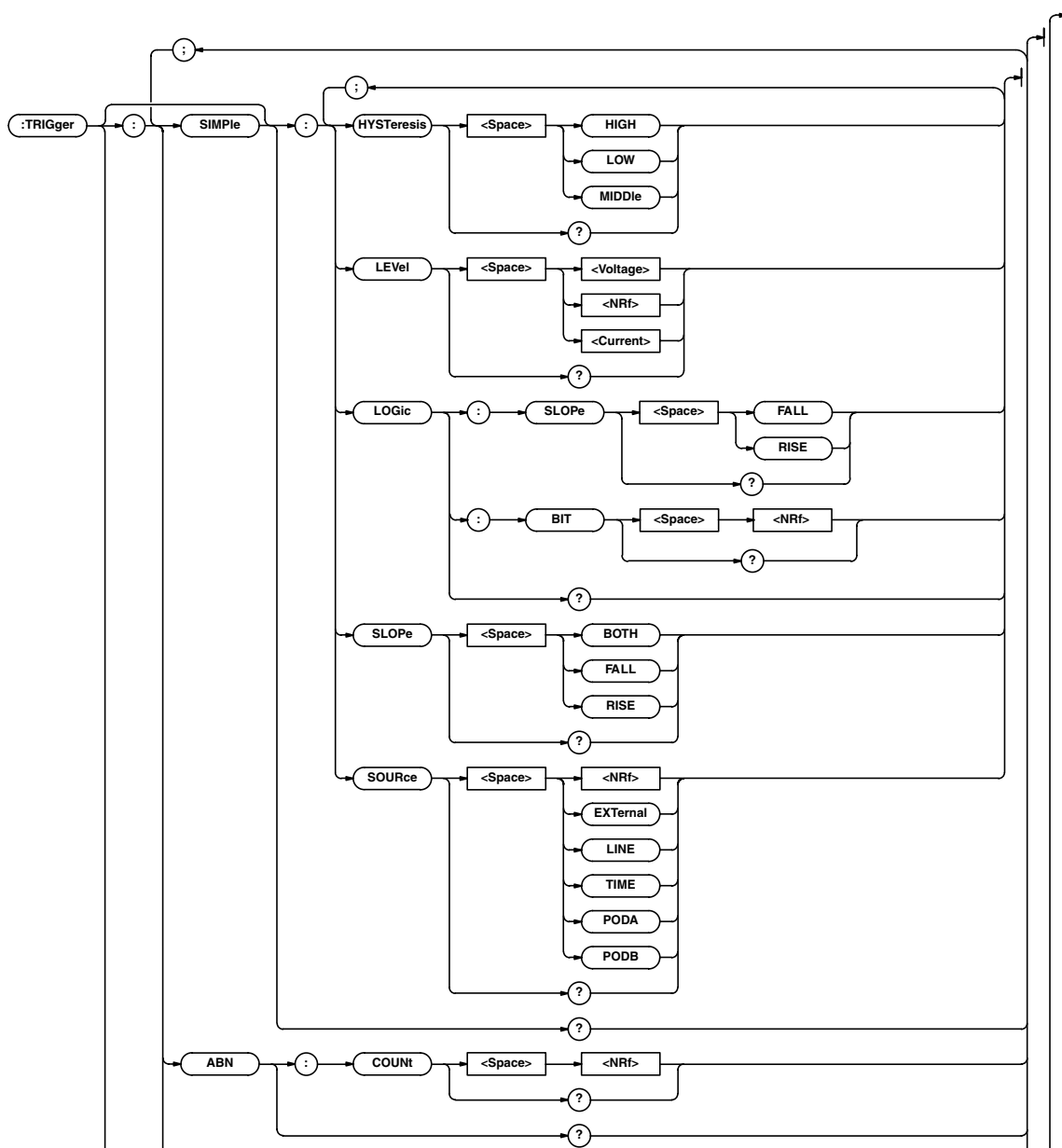
Function Sets the T/div value or queries the current setting.

Syntax :TIMEbase:TDIV {<Time> | HOUR1 | HOUR2 |  
HOUR3 | HOUR4 | HOUR5 | HOUR6 | HOUR7 | HOUR8 |  
HOUR9 | HOUR10 | HOUR12 | DAY1 | DAY2 | DAY3 }  
:TIMEbase:TDIV?

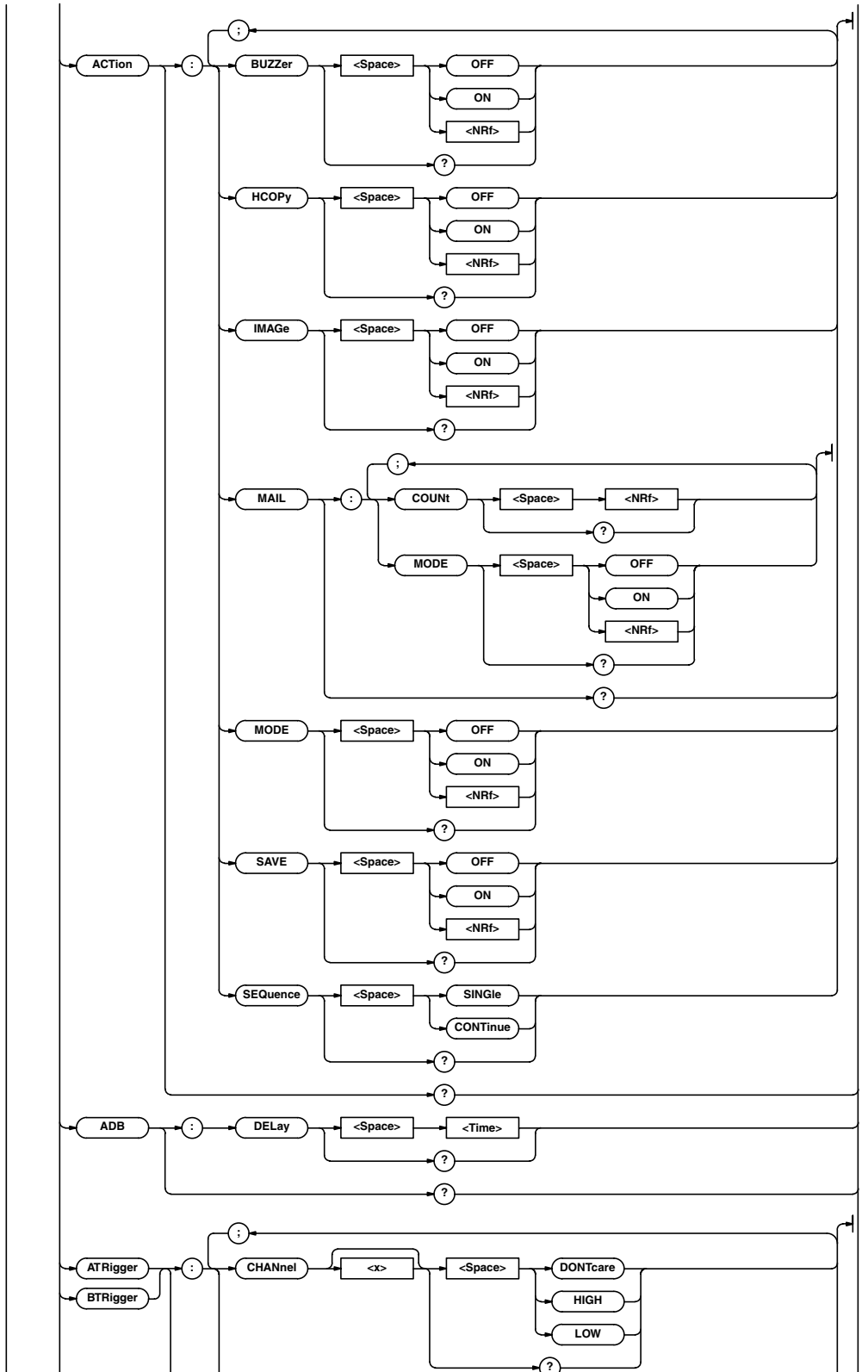
Example <Time>=500 ns to 1800 s  
:TIMEBASE:TDIV 1MS  
:TIMEBASE:TDIV? -> :TIMEBASE:  
TDIV 1.000E-03

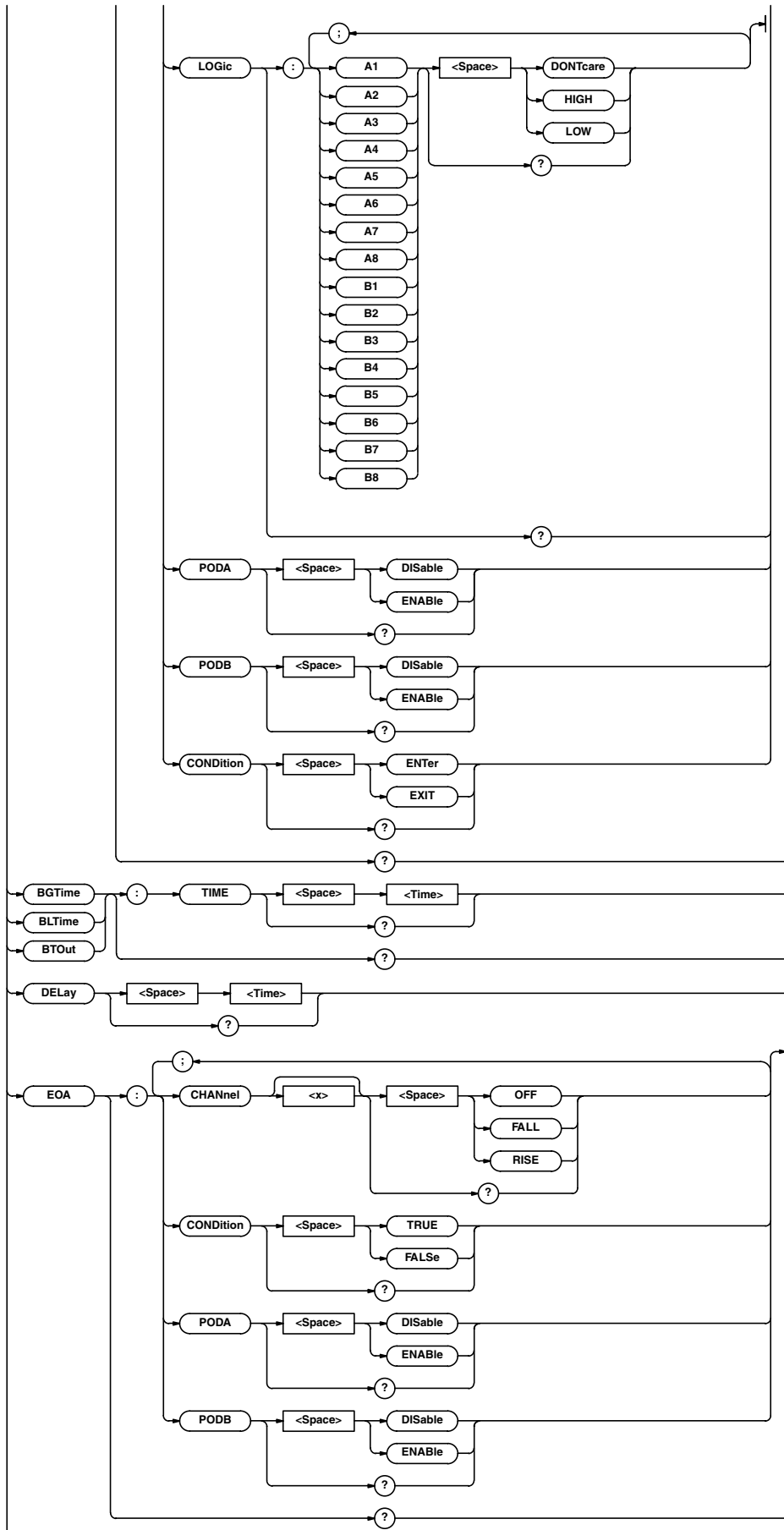
## 6.30 TRIGger Group

The commands in the trigger group deal with the trigger. You can make the same settings and inquiries as when the TRIGGER group (MODE, SIMPLE, ENHANCED, and POSITION) on the front panel is used.



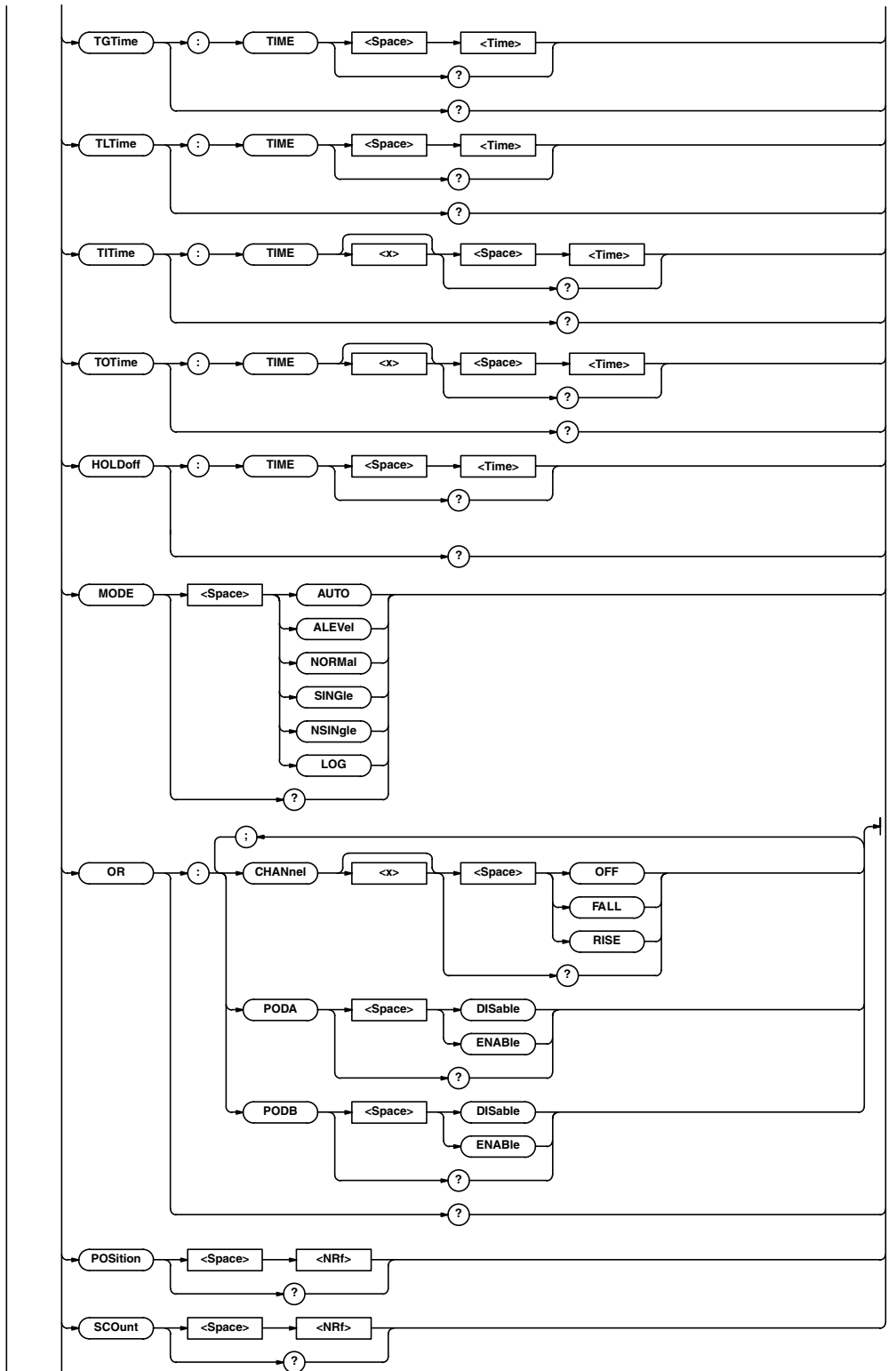
### 6.30 TRIGger Group

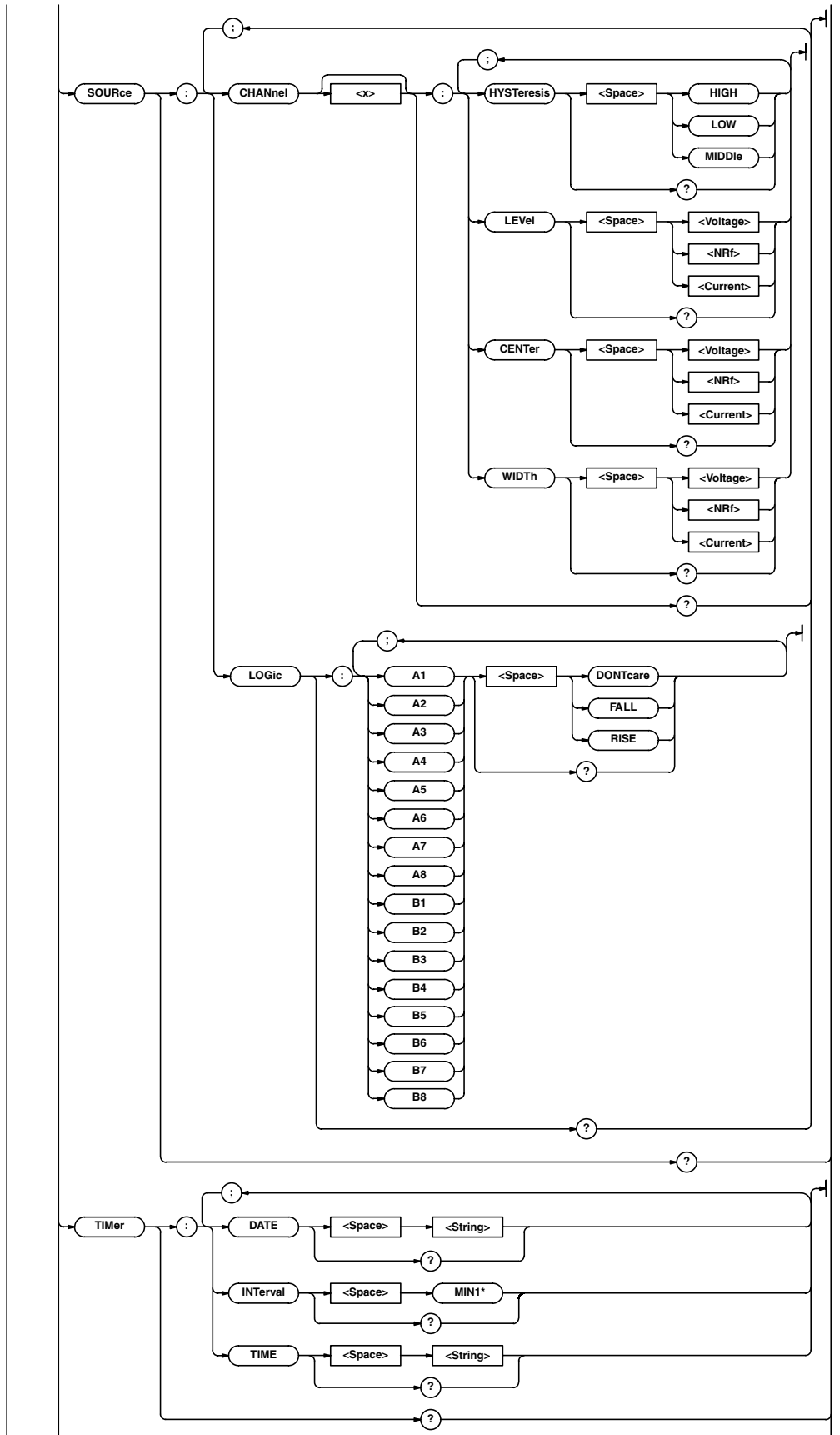




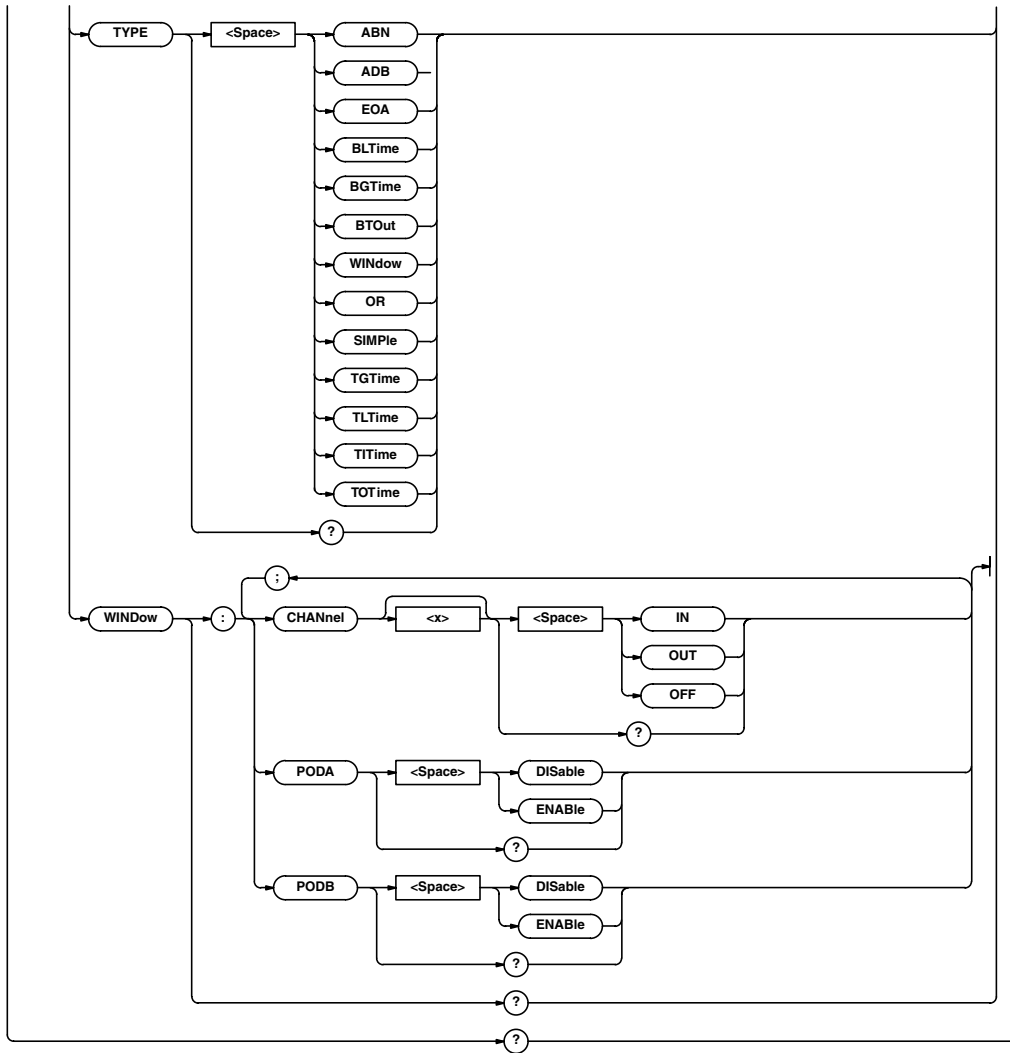


### 6.30 TRIGger Group





### 6.30 TRIGger Group



\* Select from MIN2, MIN3, MIN4, MIN5, MIN6, MIN7, MIN8, MIN9, MIN10, MIN15, MIN20, MIN25, MIN30, MIN40, MIN45, MIN50, HOUR1, HOUR2, HOUR3, HOUR4, HOUR5, HOUR6, HOUR7, HOUR8, HOUR9, HOUR10, HOUR10, HOUR12, HOUR18, and HOUR24.

**:TRIGger?**

**Function** Queries all settings related to the trigger.

**Syntax** :TRIGger?

**Example** :TRIGGER? -> :TRIGGER:ACTION:  
MODE 0;HCOPY 0;IMAGE 0;SAVE 0;  
BUZZER 0;MAIL:MODE 0;COUNT 100;:  
TRIGGER:ACTION:SEQUENCE CONTINUE;:  
TRIGGER:SIMPLE:SOURCE 1;  
LEVEL 0.0E+00;HYSTERESIS LOW;  
SLOPE RISE;LOGIC A1;:TRIGGER:ABN:  
COUNT 1;:TRIGGER:ADB:  
DELAY 0.0E+00;:TRIGGER:  
MODE AUTO;EOA:CHANNEL1 OFF;  
CHANNEL2 RISE;CHANNEL3 OFF;  
CHANNEL4 OFF;CHANNEL5 OFF;  
CHANNEL6 OFF;CHANNEL7 OFF;  
CHANNEL8 OFF;CHANNEL9 OFF;  
CHANNEL10 OFF;CHANNEL11 OFF;  
CHANNEL12 OFF;CHANNEL13 OFF;  
CHANNEL14 OFF;CHANNEL15 OFF;  
CHANNEL16 OFF;PODA DISABLE;  
PODB DISABLE;CONDITION TRUE;:  
TRIGGER:OR:CHANNEL1 RISE;  
CHANNEL2 OFF;CHANNEL3 OFF;  
CHANNEL4 OFF;CHANNEL5 OFF;  
CHANNEL6 OFF;CHANNEL7 OFF;  
CHANNEL8 OFF;CHANNEL9 OFF;  
CHANNEL10 OFF;CHANNEL11 OFF;  
CHANNEL12 OFF;CHANNEL13 OFF;  
CHANNEL14 OFF;CHANNEL15 OFF;  
CHANNEL16 OFF;PODA DISABLE;  
PODB DISABLE;:TRIGGER:TYPE SIMPLE;  
BGTIME:TIME 100.000E-09;:TRIGGER:  
BLTIME:TIME 100.000E-09;:TRIGGER:  
BTOUT:TIME 100.000E-09;:TRIGGER:  
TGTIME:TIME 100.000E-09;:TRIGGER:  
TLTIME:TIME 100.000E-09;:TRIGGER:  
TITIME:TIME1 100.000E-09;  
TIME2 200.000E-09;:TRIGGER:TOTIME:  
TIME1 100.000E-09;  
TIME2 200.000E-09;:TRIGGER:TIMER:  
DATE "2001/01/01";TIME "00:00:00";  
INTERVAL HOUR1;:TRIGGER:WINDOW:  
CHANNEL1 IN;CHANNEL2 OFF;  
CHANNEL3 OFF;CHANNEL4 OFF;  
CHANNEL5 OFF;CHANNEL6 OFF;  
CHANNEL7 OFF;CHANNEL8 OFF;  
CHANNEL9 OFF;CHANNEL10 OFF;  
CHANNEL11 OFF;CHANNEL12 OFF;  
CHANNEL13 OFF;CHANNEL14 OFF;  
CHANNEL15 OFF;CHANNEL16 OFF;  
PODA DISABLE;PODB DISABLE;:TRIGGER:  
HOLDOFF:TIME 100.000E-09;:TRIGGER:  
SOURCE:CHANNEL1:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL2:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL3:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 5.000E+00;:TRIGGER:SOURCE:  
CHANNEL4:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 5.000E+00;:TRIGGER:SOURCE:  
CHANNEL5:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 5.000E+00;:TRIGGER:SOURCE:

CHANNEL6:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 5.000E+00;:TRIGGER:SOURCE:  
CHANNEL7:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL8:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL9:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL10:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL11:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL12:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL13:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL14:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL15:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL16:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
LOGIC:A1 RISE;A2 DONTCARE;  
A3 DONTCARE;A4 DONTCARE;  
A5 DONTCARE;A6 DONTCARE;  
A7 DONTCARE;A8 DONTCARE;  
B1 DONTCARE;B2 DONTCARE;  
B3 DONTCARE;B4 DONTCARE;  
B5 DONTCARE;B6 DONTCARE;  
B7 DONTCARE;B8 DONTCARE;:TRIGGER:  
ATRIGGER:CHANNEL1 HIGH;  
CHANNEL2 DONTCARE;  
CHANNEL3 DONTCARE;  
CHANNEL4 DONTCARE;  
CHANNEL5 DONTCARE;  
CHANNEL6 DONTCARE;  
CHANNEL7 DONTCARE;  
CHANNEL8 DONTCARE;  
CHANNEL9 DONTCARE;  
CHANNEL10 DONTCARE;  
CHANNEL11 DONTCARE;  
CHANNEL12 DONTCARE;  
CHANNEL13 DONTCARE;  
CHANNEL14 DONTCARE;  
CHANNEL15 DONTCARE;  
CHANNEL16 DONTCARE;PODA DISABLE;  
PODB DISABLE;LOGIC:A1 DONTCARE;  
A2 HIGH;A3 DONTCARE;A4 DONTCARE;  
A5 DONTCARE;A6 DONTCARE;  
A7 DONTCARE;A8 DONTCARE;  
B1 DONTCARE;B2 DONTCARE;  
B3 DONTCARE;B4 DONTCARE;  
B5 DONTCARE;B6 DONTCARE;  
B7 DONTCARE;B8 DONTCARE;:TRIGGER:  
ATRIGGER:CONDITION ENTER;:TRIGGER:  
BTRIGGER:CHANNEL1 DONTCARE;  
CHANNEL2 HIGH;CHANNEL3 DONTCARE;

## 6.30 TRIGger Group

```
CHANNEL4 DONTCARE;
CHANNEL5 DONTCARE;
CHANNEL6 DONTCARE;
CHANNEL7 DONTCARE;
CHANNEL8 DONTCARE;
CHANNEL9 DONTCARE;
CHANNEL10 DONTCARE;
CHANNEL11 DONTCARE;
CHANNEL12 DONTCARE;
CHANNEL13 DONTCARE;
CHANNEL14 DONTCARE;
CHANNEL15 DONTCARE;
CHANNEL16 DONTCARE;PODA DISABLE;
PODB DISABLE;LOGIC:A1 HIGH;
A2 DONTCARE;A3 DONTCARE;
A4 DONTCARE;A5 DONTCARE;
A6 DONTCARE;A7 DONTCARE;
A8 DONTCARE;B1 DONTCARE;
B2 DONTCARE;B3 DONTCARE;
B4 DONTCARE;B5 DONTCARE;
B6 DONTCARE;B7 DONTCARE;
B8 DONTCARE;;TRIGGER:BTRIGGER:
CONDITION ENTER;;TRIGGER:
DELAY 0.0E+00;POSITION 50.000;
SCOUNT 2
```

### **:TRIGger:ABN? (A -> B(n))**

**Function** Queries all settings related to the A!s!ÆB(n) trigger.

**Syntax** :TRIGger:ABN?

**Example** :TRIGGER:ABN? -> :TRIGGER:ABN:COUNT 1

### **:TRIGger:ABN:COUNT**

**Function** Sets the number times condition B is to be satisfied for the A -> B(n) trigger or queries the current setting.

**Syntax** :TRIGger:ABN:COUNT {<Nrf>}  
:TRIGger:ABN:COUNT?  
<Nrf>=1 to 255

**Example** :TRIGGER:ABN:COUNT 1  
:TRIGGER:ABN:COUNT? -> :TRIGGER:  
ABN:COUNT 1

### **:TRIGger:ACTion?**

**Function** Queries all settings related to the action-on-trigger.

**Syntax** :TRIGger:ACTion?

**Example** :TRIGGER:ACTION? -> :TRIGGER:  
ACTION:MODE 0;HCOPY 0;IMAGE 0;  
SAVE 0;BUZZER 0;MAIL:MODE 0;  
COUNT 100;;TRIGGER:ACTION:  
SEQUENCE CONTINUE

### **:TRIGger:ACTion:BUZZer**

**Function** Sets whether a beep is sounded when a action is activated or queries the current setting.

**Syntax** :TRIGger:ACTion:BUZZer {<Boolean>}  
:TRIGger:ACTion:BUZZer?

**Example** :TRIGGER:ACTION:BUZZER ON  
:TRIGGER:ACTION:BUZZER? ->  
:TRIGGER:ACTION:BUZZER 1

### **:TRIGger:ACTion:HCOPY**

**Function** Sets whether to output a hard copy to the built-in printer or an external printer when an action is activated, or queries the current setting.

**Syntax** :TRIGger:ACTion:HCOPY {<Boolean>}  
:TRIGger:ACTion:HCOPY?

**Example** :TRIGGER:ACTION:HCOPY ON  
:TRIGGER:ACTION:HCOPY? -> :TRIGGER:  
ACTION:HCOPY 1

### **:TRIGger:ACTion:IMAGE**

**Function** Sets whether or not to save the screen image to the external storage medium when action is activated or queries the current setting.

**Syntax** :TRIGger:ACTion:IMAGE {<Boolean>}  
:TRIGger:ACTion:IMAGE?

**Example** :TRIGGER:ACTION:IMAGE ON  
:TRIGGER:ACTION:IMAGE? -> :TRIGGER:  
ACTION:IMAGE 1

### **:TRIGger:ACTion:MAIL?**

**Function** Sets whether to save the screen image to the storage medium when an action is activated or queries the current setting.

**Syntax** :TRIGger:ACTion:MAIL?

**Example** :TRIGGER:ACTION:MAIL? -> :TRIGGER:  
ACTION:MAIL:MODE 0;COUNT 100

**Description** Can be used when the Ethernet interface (option) is installed.

### **:TRIGger:ACTion:MAIL:COUNT**

**Function** Sets the upper limit on the number of transmitted mail when an action is activated or queries the current setting.

**Syntax** :TRIGger:ACTion:MAIL:COUNT {<Nrf>}  
:TRIGger:ACTion:MAIL:COUNT?  
<Nrf>=1 to 1000

**Example** :TRIGGER:ACTION:MAIL:COUNT 100  
:TRIGGER:ACTION:MAIL:COUNT? ->  
:TRIGGER:ACTION:MAIL:COUNT 100

**Description** Can be used when the Ethernet interface (option) is installed.

### **:TRIGger:ACTion:MAIL:MODE**

**Function** Queries whether a mail is set to be transmitted when an action is activated.

**Syntax** :TRIGger:ACTion:MAIL:MODE  
{<Boolean>}  
:TRIGger:ACTion:MAIL:MODE?

**Example** :TRIGGER:ACTION:MAIL:MODE OFF  
:TRIGGER:ACTION:MAIL:MODE? ->  
:TRIGGER:ACTION:MAIL:MODE 0

**Description** Can be used when the Ethernet interface (option) is installed.

### **:TRIGger:ACTion:MODE**

**Function** Turns ON/OFF the action-on-trigger or queries the current setting.

**Syntax** :TRIGger:ACTion:MODE {<Boolean>}  
:TRIGger:ACTion:MODE?

**Example** :TRIGGER:ACTION:MODE ON  
:TRIGGER:ACTION:MODE? -> :TRIGGER:  
ACTION:MODE 1

**:TRIGger:ACTion:SAVE**

Function Sets whether or not to save the waveform data to the storage medium (ON/OFF) when an action is activated, or queries the current setting.

Syntax :TRIGger:ACTion:SAVE {<Boolean>}  
:TRIGger:ACTion:SAVE?

Example :TRIGGER:ACTION:SAVE ON  
:TRIGGER:ACTION:SAVE? -> :TRIGGER:  
ACTION:SAVE 1

**:TRIGger:ACTion:SEQuence**

Function Sets whether to repeat the action or queries the current setting.

Syntax :TRIGger:ACTion:SEQuence  
{CONTInue|SINGle}  
:TRIGger:ACTion:SEQuence?

Example :TRIGGER:ACTION:SEQUENCE CONTINUE  
:TRIGGER:ACTION:SEQUENCE? ->  
:TRIGGER:ACTION:SEQUENCE CONTINUE

**:TRIGger:ADB? (A Delay B)**

Function Queries all settings related to the A Delay B trigger.

Syntax :TRIGger:ADB?

Example :TRIGGER:ADB? -> :TRIGGER:ADB:  
DELAY 0.0E+00

**:TRIGger:ADB:DELAy**

Function Sets the delay time of condition B for the A Delay B trigger or queries the current setting.

Syntax :TRIGger:ADB:DELAy {<Time>}  
:TRIGger:ADB:DELAy?  
<Time>=0 ns to 10 s

Example :TRIGGER:ADB:DELAY 1US  
:TRIGGER:ADB:DELAY? -> :TRIGGER:  
ADB:DELAY 1.000000E-06

**:TRIGger:ATRigger?**

Function Queries all settings related to condition A.

Syntax :TRIGger:ATRigger?

Example :TRIGGER:ATRIGGER? -> :TRIGGER:  
ATRIGGER:CHANNEL1 HIGH;  
CHANNEL2 DONTCARE;  
CHANNEL3 DONTCARE;  
CHANNEL4 DONTCARE;  
CHANNEL5 DONTCARE;  
CHANNEL6 DONTCARE;  
CHANNEL7 DONTCARE;  
CHANNEL8 DONTCARE;  
CHANNEL9 DONTCARE;  
CHANNEL10 DONTCARE;  
CHANNEL11 DONTCARE;  
CHANNEL12 DONTCARE;  
CHANNEL13 DONTCARE;  
CHANNEL14 DONTCARE;  
CHANNEL15 DONTCARE;  
CHANNEL16 DONTCARE;PODA DISABLE;  
PODB DISABLE;LOGIC:A1 DONTCARE;  
A2 HIGH;A3 DONTCARE;A4 DONTCARE;  
A5 DONTCARE;A6 DONTCARE;  
A7 DONTCARE;A8 DONTCARE;  
B1 DONTCARE;B2 DONTCARE;  
B3 DONTCARE;B4 DONTCARE;  
B5 DONTCARE;B6 DONTCARE;  
B7 DONTCARE;B8 DONTCARE; :  
TRIGGER:ATRIGGER:CONDITION ENTER

**:TRIGger:ATRigger:CHANnel<x>**

Function Sets the channel condition of condition A or queries the current setting.

Syntax :TRIGger:ATRigger:CHANnel<x>  
{DONTcare|HIGH|LOW}  
:TRIGger:ATRigger:CHANnel<x>?  
<x>=1 to 16

Example (The following is an example for CH1.)  
:TRIGGER:ATRIGGER:CHANNEL1 HIGH  
:TRIGGER:ATRIGGER:CHANNEL1? ->  
:TRIGGER:ATRIGGER:CHANNEL1 HIGH

**:TRIGger:ATRigger:LOGic?**

Function Queries all settings related the logic of condition A.

Syntax :TRIGger:ATRigger:LOGic?

Example :TRIGGER:ATRIGGER:LOGIC? ->  
:TRIGGER:  
ATRIGGER:LOGIC:A1 DONTCARE;A2 HIGH;  
A3 DONTCARE;A4 DONTCARE;  
A5 DONTCARE;A6 DONTCARE;  
A7 DONTCARE;A8 DONTCARE;  
B1 DONTCARE;B2 DONTCARE;  
B3 DONTCARE;B4 DONTCARE;  
B5 DONTCARE;B6 DONTCARE;  
B7 DONTCARE;B8 DONTCARE

**:TRIGger:ATRigger:LOGic:****{A1~A8 | B1~B8}**

Function Sets the logic bit of condition A or queries the current setting.

Syntax :TRIGger:ATRigger:LOGic:{A1~A8 |  
B1~B8} {DONTcare|HIGH|LOW}  
:TRIGger:ATRigger:LOGic:{A1~A8 |  
B1~B8}?

Example :TRIGGER:ATRIGGER:LOGIC:A1 HIGH  
:TRIGGER:ATRIGGER:LOGIC:A1? ->  
:TRIGGER:ATRIGGER:LOGIC:A1 HIGH

**:TRIGger:ATRigger:{PODA | POdB}**

Function Sets the logic condition of condition A or queries the current setting.

Syntax :TRIGger:ATRigger:{PODA|POdB}  
{DISable|ENABle}  
:TRIGger:ATRigger:{PODA|POdB}?

Example :TRIGGER:ATRIGGER:PODA DISABLE  
:TRIGGER:ATRIGGER:PODA? ->  
:TRIGGER:ATRIGGER:PODA DISABLE

**:TRIGger:ATRigger:CONDition**

Function Sets the condition to be satisfied for condition A or queries the current setting.

Syntax :TRIGger:ATRigger:CONDition  
{ENTer|EXIT}  
:TRIGger:ATRigger:CONDition?

Example :TRIGGER:ATRIGGER:CONDITION ENTER  
:TRIGGER:ATRIGGER:CONDITION? ->  
:TRIGGER:ATRIGGER:CONDITION ENTER

Description This command is valid when the trigger type is set to ABN or ADB.

## 6.30 TRIGger Group

### **:TRIGger:BGTime?**

Function Queries all settings related to the B>Time trigger.

Syntax :TRIGger:BGTime?

Example :TRIGGER:BGTIME? -> :TRIGGER:  
BGTIME:TIME 100.000E-09

### **:TRIGger:BGTime:TIME**

Function Sets the pulse width of the B>Time trigger or queries the current setting.

Syntax :TRIGger:BGTime:TIME {<Time>}  
:TRIGger:BGTime:TIME?  
<Time>=100 ns to 10 s

Example :TRIGGER:BGTIME:TIME 200NS  
:TRIGGER:BGTIME:TIME? -> :TRIGGER:  
BGTIME:TIME 200.000E-09

### **:TRIGger:BLTime?**

Function Queries all settings related to the B<Time trigger.

Syntax :TRIGger:BLTime?

Example :TRIGGER:BLTIME? -> :TRIGGER:  
BLTIME:TIME 200.000E-09

### **:TRIGger:BLTIME:TIME**

Function Sets the pulse width of the B<Time trigger or queries the current setting.

Syntax :TRIGger:BLTime:TIME {<Time>}  
:TRIGger:BLTime:TIME?  
<Time>=100 ns to 10 s

Example :TRIGGER:BLTIME:TIME 600NS  
:TRIGGER:BLTIME:TIME? -> :TRIGGER:  
BLTIME:TIME 600.000E-09

### **:TRIGger:BTOut?**

Function Queries all settings related to the B Time Out trigger.

Syntax :TRIGger:BTOut?

Example :TRIGGER:BTOUT? -> :TRIGGER:BTOUT:  
TIME 600.000E-09

### **:TRIGger:BTOut:TIME**

Function Sets the pulse width of the B Time Out trigger or queries the current setting.

Syntax :TRIGger:BTOut:TIME {<Time>}  
:TRIGger:BTOut:TIME?  
<Time>=100 ns to 10 s

Example :TRIGGER:BTOUT:TIME 8S  
:TRIGGER:BTOUT:TIME? -> :TRIGGER:  
BTOUT:TIME 8.000000000000E+00

### **:TRIGger:BTRigger?**

Function Queries all settings related to condition B.

Syntax :TRIGger:BTRigger?

Example :TRIGGER:BTRIGGER? -> :TRIGGER:  
BTRIGGER:CHANNEL1 DONTCARE;  
CHANNEL2 HIGH;CHANNEL3 DONTCARE;  
CHANNEL4 DONTCARE;  
CHANNEL5 DONTCARE;  
CHANNEL6 DONTCARE;  
CHANNEL7 DONTCARE;  
CHANNEL8 DONTCARE;  
CHANNEL9 DONTCARE;  
CHANNEL10 DONTCARE;  
CHANNEL11 DONTCARE;  
CHANNEL12 DONTCARE;  
CHANNEL13 DONTCARE;  
CHANNEL14 DONTCARE;  
CHANNEL15 DONTCARE;  
CHANNEL16 DONTCARE;PODA DISABLE;  
PODB DISABLE;LOGIC:A1 HIGH;  
A2 DONTCARE;A3 DONTCARE;  
A4 DONTCARE;A5 DONTCARE;  
A6 DONTCARE;A7 DONTCARE;  
A8 DONTCARE;B1 DONTCARE;  
B2 DONTCARE;B3 DONTCARE;  
B4 DONTCARE;B5 DONTCARE;  
B6 DONTCARE;B7 DONTCARE;  
B8 DONTCARE;:TRIGGER:BTRIGGER:  
CONDITION ENTER

### **:TRIGger:BTRigger:CHANnel<x>**

Function Sets the channel condition of condition B or queries the current setting.

Syntax :TRIGger:BTRigger:CHANnel<x>  
{DONTcare|HIGH|LOW}  
:TRIGger:BTRigger:CHANnel<x>?  
<x>=1 to 16

Example (The following is an example for CH1.)  
:TRIGGER:BTRIGGER:CHANNEL1 HIGH  
:TRIGGER:BTRIGGER:CHANNEL1? ->  
:TRIGGER:BTRIGGER:CHANNEL1 HIGH

### **:TRIGger:BTRigger:LOGic?**

Function Queries all settings related the logic of condition B.

Syntax :TRIGger:BTRigger:LOGic?

Example :TRIGGER:BTRIGGER:LOGIC? ->  
:TRIGGER:BTRIGGER:LOGIC:A1 HIGH;  
A2 DONTCARE;A3 DONTCARE;  
A4 DONTCARE;A5 DONTCARE;  
A6 DONTCARE;A7 DONTCARE;  
A8 DONTCARE;B1 DONTCARE;  
B2 DONTCARE;B3 DONTCARE;  
B4 DONTCARE;B5 DONTCARE;  
B6 DONTCARE;B7 DONTCARE;B8 DONTCARE

### **:TRIGger:BTRigger:LOGic:**

**{A1~A8 | B1~B8}**

Function Sets the logic bit of condition B or queries the current setting.

Syntax :TRIGger:BTRigger:LOGic:{A1~A8 |  
B1~B8} {DONTcare|HIGH|LOW}  
:TRIGger:BTRigger:LOGic:{A1~A8 |  
B1~B8}?

Example :TRIGGER:BTRIGGER:LOGIC:B1 HIGH  
:TRIGGER:BTRIGGER:LOGIC:B1? ->  
:TRIGGER:BTRIGGER:LOGIC:B1 HIGH

**:TRIGger:BTRigger:{PODA|PODB}**

Function Sets the logic condition of condition B or queries the current setting.

Syntax :TRIGger:BTRigger:{PODA|PODB}  
{DISable|ENABLE}

Example :TRIGger:BTRigger:{PODA|PODB}?  
:TRIGGER:BTRIGGER:PODB ENABLE  
:TRIGGER:BTRIGGER:PODB? ->  
:TRIGGER:BTRIGGER:PODB ENABLE

**:TRIGger:BTRigger:CONDition**

Function Sets the condition to be satisfied for condition B or queries the current setting.

Syntax :TRIGger:BTRigger:CONDition  
{ENTer|EXIT}

Example :TRIGger:BTRigger:CONDition?  
:TRIGGER:BTRIGGER:CONDITION? ->  
:TRIGGER:BTRIGGER:CONDITION ENTER

Description This command is valid when the trigger type is set to ABN or ADB.

**:TRIGger:DElay**

Function Sets the delay (time from the trigger point to the trigger position) or queries the current setting.

Syntax :TRIGger:DElay {<Time>}  
:TRIGger:DElay?

Example :TRIGGER:DElay 2US  
:TRIGGER:DElay? -> :TRIGGER:  
DElay 2.000000E-06

Description

- Fixed to 0 when the time base clock is set to external clock.
- On the DL750, the trigger delay sets the time from the trigger point to the trigger position. To set or query the trigger position, use :TRIGger:POsition.

**:TRIGger:EOA?**

Function Queries all settings related to the Edge on A trigger.

Syntax :TRIGger:EOA?

Example :TRIGGER:EOA? -> :TRIGGER:EOA:  
CHANNEL1 OFF;CHANNEL2 RISE;  
CHANNEL3 OFF;CHANNEL4 OFF;  
CHANNEL5 OFF;CHANNEL6 OFF;  
CHANNEL7 OFF;CHANNEL8 OFF;  
CHANNEL9 OFF;CHANNEL10 OFF;  
CHANNEL11 OFF;CHANNEL12 OFF;  
CHANNEL13 OFF;CHANNEL14 OFF;  
CHANNEL15 OFF;CHANNEL16 OFF;  
PODA DISABLE;PODB DISABLE;  
CONDITION TRUE

**:TRIGger:EOA:CHANnel<x>**

Function Sets the channel condition of the Edge on A trigger or queries the current setting.

Syntax :TRIGger:EOA:CHANnel<x>  
{OFF|FALL|RISE}

Example :TRIGger:EOA:CHANnel<x>?  
<x>=1 to 16  
(The following is an example for CH1.)  
:TRIGGER:EOA:CHANNEL1 FALL  
:TRIGGER:EOA:CHANNEL1? -> :TRIGGER:  
EOA:CHANNEL1 FALL

**:TRIGger:EOA:{PODA|PODB}**

Function Sets the logic condition of the Edge on A trigger or queries the current setting.

Syntax :TRIGger:EOA:{PODA|PODB}  
{DISable|ENABLE}

Example :TRIGger:EOA:{PODA|PODB}?  
:TRIGGER:EOA:PODB ENABLE  
:TRIGGER:EOA:PODB? -> :TRIGGER:EOA:  
PODB ENABLE

**:TRIGger:EOA:CONDition**

Function Sets the condition to be satisfied for the Edge on A trigger or queries the current setting.

Syntax :TRIGger:EOA:CONDition {TRUE|FALSE}

Example :TRIGger:EOA:CONDition?  
:TRIGGER:EOA:CONDITION TRUE  
:TRIGGER:EOA:CONDITION? ->  
:TRIGGER:EOA:CONDITION TRUE

**:TRIGger:HOLDoff?**

Function Queries all settings related to the hold off.

Syntax :TRIGger:HOLDoff?

Example :TRIGGER:HOLDOFF? -> :TRIGGER:  
HOLDOFF:TIME 100.000E-09

**:TRIGger:HOLDoff:TIME**

Function Sets the hold off time or queries the current setting.

Syntax :TRIGger:HOLDoff:TIME {<Time>}  
:TRIGger:HOLDoff:TIME?

Example :TRIGGER:HOLDOFF:TIME 500NS  
:TRIGGER:HOLDOFF:TIME? -> :TRIGGER:  
HOLDOFF:TIME 500.000E-09

**:TRIGger:MODE**

Function Sets the trigger mode or queries the current setting.

Syntax :TRIGger:MODE  
{AUTO|ALEvel|NORMal|SINGle|NSINGle|LOG}

Example :TRIGger:MODE AUTO  
:TRIGGER:MODE? -> :TRIGGER:  
MODE AUTO

Description To set or query the count when NSINGle is specified, use :TRIGger:SCount.

**:TRIGger:OR?**

Function Queries all settings related to the OR trigger.

Syntax :TRIGger:OR?

Example :TRIGGER:OR? -> :TRIGGER:OR:  
CHANNEL1 RISE;CHANNEL2 OFF;  
CHANNEL3 OFF;CHANNEL4 OFF;  
CHANNEL5 OFF;CHANNEL6 OFF;  
CHANNEL7 OFF;CHANNEL8 OFF;  
CHANNEL9 OFF;CHANNEL10 OFF;  
CHANNEL11 OFF;CHANNEL12 OFF;  
CHANNEL13 OFF;CHANNEL14 OFF;  
CHANNEL15 OFF;CHANNEL16 OFF;  
PODA DISABLE;PODB DISABLE



## 6.30 TRIGger Group

### **:TRIGger:OR:CHANnel<x>**

**Function** Sets the channel condition of the OR trigger or queries the current setting.

**Syntax**  
:TRIGger:OR:CHANnel<x>  
{OFF|FALL|RISE}  
:TRIGger:OR:CHANnel<x>?  
<x>=1 to 16

**Example** (The following is an example for CH1.)  
:TRIGGER:OR:CHANNEL1 RISE  
:TRIGGER:OR:CHANNEL1? -> :TRIGGER:  
OR:CHANNEL1 RISE

### **:TRIGger:OR:{PODA|PODB}**

**Function** Sets the logic condition of the OR trigger or queries the current setting.

**Syntax**  
:TRIGger:OR:{PODA|PODB}  
{DISable|ENABLE}  
:TRIGger:OR:{PODA|PODB}?

**Example**  
:TRIGGER:OR:PODA ENABLE  
:TRIGGER:OR:PODA? -> :TRIGGER:OR:  
PODA ENABLE

### **:TRIGger:POSition**

**Function** Sets the trigger position or queries the current setting.

**Syntax**  
:TRIGger:POSition {<NRf>}  
:TRIGger:POSition?  
<NRf>=0 to 100 (% , 0.1 steps)

**Example**  
:TRIGGER:POSITION 50  
:TRIGGER:POSITION? -> :TRIGGER:  
POSITION 50.000

### **:TRIGger:SCount (Single(N) Count)**

**Function** Sets the number of times the trigger is to be activated when the trigger mode is Single(N) or queries the current setting.

**Syntax**  
:TRIGger:SCount {<NRf>}  
:TRIGger:SCount?  
<NRf>=1 to 2000

**Example**  
:TRIGGER:SCOUNT 100  
:TRIGGER:SCOUNT? -> :TRIGGER:  
SCOUNT 100

### **:TRIGger[:SIMPlE]?**

**Function** Queries all settings related to the simple trigger.

**Syntax**  
:TRIGger:SIMPlE?  
**Example**  
:TRIGGER:SIMPLE? -> :TRIGGER:  
SIMPLE:SOURCE 1;LEVEL 0.0E+00;  
HYSTERESIS LOW;SLOPE RISE;LOGIC:  
BIT 2;SLOPE RISE

### **:TRIGger[:SIMPlE]:HYSTEResis**

**Function** Sets the hysteresis of the simple trigger search level or queries the current setting.

**Syntax**  
:TRIGger[:SIMPlE]:HYSTEResis  
{HIGH|LOW|MIDDLE}  
:TRIGger[:SIMPlE]:HYSTEResis?

**Example**  
:TRIGGER:SIMPLE:HYSTERESIS MIDDLE  
:TRIGGER:SIMPLE:HYSTERESIS? ->  
:TRIGGER:SIMPLE:HYSTERESIS MIDDLE

**Description** This command cannot be used when the trigger is set to EXTERNAL, LINE, TIME, LOGICA, or LOGICB.

### **:TRIGger[:SIMPlE]:LEVEl**

**Function** Sets the trigger level of the channel specified using TRIGger[:SIMPlE]:  
SOURce for the simple trigger or queries the current setting.

**Syntax**  
:TRIGger[:SIMPlE]:LEVEl  
{<Voltage>|<NRf>|<Current>}  
:TRIGger[:SIMPlE]:LEVEl?  
{<Voltage>|<NRf>|<Current>}=The selectable  
range varies depending on the range and  
offset settings. For details, see the *DL750  
User's Manual*.

**Example**  
:TRIGGER:SIMPLE:LEVEL 0V  
:TRIGGER:SIMPLE:LEVEL? -> :TRIGGER:  
SIMPLE:LEVEL 0.0E+00

**Description** This command cannot be used when the trigger is set to EXTERNAL, LINE, TIME, LOGICA, or LOGICB.

### **:TRIGger[:SIMPlE]:SLOPe**

**Function** Sets the trigger slope of the channel specified using TRIGger[:SIMPlE]:  
SOURce for the simple trigger or queries the current setting.

**Syntax**  
:TRIGger[:SIMPlE]:SLOPe  
{BOTH|FALL|RISE}  
:TRIGger[:SIMPlE]:SLOPe?

**Example**  
:TRIGGER:SIMPLE:SLOPE RISE  
:TRIGGER:SIMPLE:SLOPE? -> :TRIGGER:  
SIMPLE:SLOPE RISE

**Description** This command cannot be used when the trigger is set to EXTERNAL, LINE, TIME, LOGICA, or LOGICB.

### **:TRIGger:SIMPlE:SOURce**

**Function** Sets the trigger source of the simple trigger search level or queries the current setting.

**Syntax**  
:TRIGger[:SIMPlE]:SOURce  
{<NRf>|EXTERNAL|LINE|TIME|PODA|PODB}  
:TRIGger[:SIMPlE]:SOURce?  
<NRf>=1 to 16

**Example**  
:TRIGGER:SIMPLE:SOURCE 1  
:TRIGGER:SIMPLE:SOURCE? ->  
:TRIGGER:SIMPLE:SOURCE 1

**Description** Of the :TRIGger:SIMPlE commands,  
:SIMPlE cannot be abbreviated for the  
:TRIGger:SIMPlE:SOURce command.

### **:TRIGger[:SIMPlE]:LOGic?**

**Function** Queries all settings related to the logic of the simple trigger.

**Syntax**  
:TRIGger[:SIMPlE]:LOGic?  
**Example**  
:TRIGGER:SIMPLE:LOGIC? -> :TRIGGER:  
SIMPLE:LOGIC:BIT 2;SLOPE RISE

### **:TRIGger[:SIMPlE]:LOGic:BIT**

**Function** Sets the bit for setting the trigger slope or queries the current setting.

**Syntax**  
:TRIGger[:SIMPlE]:LOGic:BIT {<NRf>}  
:TRIGger[:SIMPlE]:LOGic:BIT?  
<NRf>=1 to 8

**Example**  
:TRIGGER:SIMPLE:LOGIC:BIT 2  
:TRIGGER:SIMPLE:LOGIC:BIT? ->  
:TRIGGER:SIMPLE:LOGIC:BIT 2

**:TRIGger[:SIMPlE]:LOGic:SLOPe**

**Function** Sets the logic trigger slope or queries the current setting.

**Syntax** :TRIGger[:SIMPlE]:LOGic:SLOPe  
{FALL|RISE}

**Example** :TRIGGER:SIMPLE:LOGIC:SLOPE FALL  
:TRIGGER:SIMPLE:LOGIC:SLOPE? ->  
:TRIGGER:SIMPLE:LOGIC:SLOPE FALL

**:TRIGger:SOURce?**

**Function** Queries all settings related to the trigger source of the enhanced trigger.

**Syntax** :TRIGger:SOURce?

**Example** :TRIGGER:SOURCE? -> :TRIGGER:  
SOURCE:CHANNEL1:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL2:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL3:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL4:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL5:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL6:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 5.000E+00;:TRIGGER:SOURCE:  
CHANNEL7:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL8:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL9:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL10:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL11:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL12:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL13:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL14:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 50.000E+00;:TRIGGER:SOURCE:  
CHANNEL15:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 5.000E+00;:TRIGGER:SOURCE:  
CHANNEL16:LEVEL 0.0E+00;  
HYSTERESIS LOW;CENTER 0.0E+00;  
WIDTH 5.000E+00;:TRIGGER:SOURCE:  
LOGIC:A1 RISE;A2 DONTCARE;  
A3 DONTCARE;A4 DONTCARE;  
A5 DONTCARE;A6 DONTCARE;  
A7 DONTCARE;A8 DONTCARE;

B1 DONTCARE;B2 DONTCARE;  
B3 DONTCARE;B4 DONTCARE;  
B5 DONTCARE;B6 DONTCARE;  
B7 DONTCARE;B8 DONTCARE

**:TRIGger:SOURce:CHANnel<x>?**

**Function** Queries all settings related to the trigger source of the specified channel of the enhanced trigger.

**Syntax** :TRIGger:SOURce:CHANnel<x>?  
<x>=1 to 16

**Example** (The following is an example for CH1.)

:TRIGGER:SOURCE:CHANNEL1? ->  
:TRIGGER:SOURCE:CHANNEL1:  
LEVEL 0.0E+00;HYSTERESIS LOW;  
CENTER 0.0E+00;WIDTH 50.000E+00

**:TRIGger:SOURce:CHANnel<x>:CENTER**

**Function** Sets the center of the window trigger or queries the current setting.

**Syntax** :TRIGger:SOURce:CHANnel<x>:CENTER  
{<Voltage>|<NRf>|<Current>}  
:TRIGger:SOURce:CHANnel<x>:CENTER?  
<x>=1 to 16

{<Voltage>|<NRf>|<Current>}=The selectable range varies depending on the range and offset settings. For details, see the *DL750 User's Manual*.

**Example** (The following is an example for CH1.)

:TRIGGER:SOURCE:CHANNEL1:CENTER 5V  
:TRIGGER:SOURCE:CHANNEL1:CENTER? ->  
:TRIGGER:SOURCE:CHANNEL1:  
CENTER 5.000E+00

**:TRIGger:SOURce:CHANnel<x>****:HYSTeresis**

**Function** Sets the hysteresis of the enhanced trigger search level or queries the current setting.

**Syntax** :TRIGger:SOURce:CHANnel<x>:  
HYSTeresis {HIGH|LOW|MIDDLE}  
:TRIGger:SOURce:CHANnel<x>:  
HYSTeresis?  
<x>=1 to 16

**Example** (The following is an example for CH1.)

:TRIGGER:SOURCE:CHANNEL1:  
HYSTERESIS HIGH  
:TRIGGER:SOURCE:CHANNEL1:  
HYSTERESIS? -> :TRIGGER:SOURCE:  
CHANNEL1:HYSTERESIS HIGH

**:TRIGger:SOURce:CHANnel<x>:LEVEL**

**Function** Sets the trigger level of the specified channel for the enhanced trigger or queries the current setting.

**Syntax** :TRIGger:SOURce:CHANnel<x>:LEVEL  
{<Voltage>|<NRf>|<Current>}  
:TRIGger:SOURce:CHANnel<x>:LEVEL?  
<x>=1 to 16

{<Voltage>|<NRf>|<Current>}=The selectable range varies depending on the range and offset settings. For details, see the *DL750 User's Manual*.

**Example** (The following is an example for CH1.)

:TRIGGER:SOURCE:CHANNEL1:LEVEL 0V  
:TRIGGER:SOURCE:CHANNEL1:LEVEL? ->  
:TRIGGER:SOURCE:CHANNEL1:  
LEVEL 0.0E+00

## 6.30 TRIGger Group

### **:TRIGger:SOURce:CHANnel<x>:WIDTh**

**Function** Sets the width of the window trigger or queries the current setting.

**Syntax** :TRIGger:SOURce:CHANnel<x>:WIDTh {<Voltage>|<NRf>|<Current>}  
:TRIGger:SOURce:CHANnel<x>:WIDTh? <x>=1 to 16  
{<Voltage>|<NRf>|<Current>}=The selectable range varies depending on the range and offset settings. For details, see the *DL750 User's Manual*.

**Example** (The following is an example for CH1.)  
:TRIGGER:SOURCE:CHANNEL1:WIDTH 55V  
:TRIGGER:SOURCE:CHANNEL1:WIDTH? ->  
:TRIGGER:SOURCE:CHANNEL1:  
WIDTH 55.000E+00

### **:TRIGger:SOURce:LOGic:{A1~A8|B1~B8}**

**Function** Sets the logic bit condition or queries the current setting.

**Syntax** :TRIGger:SOURce:LOGic:{A1~A8|B1~B8} {DONTcare|FALL|RISE}  
:TRIGger:SOURce:LOGic:{A1~A8|B1~B8}? <x>=1 to 16

**Example** :TRIGGER:SOURCE:LOGIC:A1 RISE  
:TRIGGER:SOURCE:LOGIC:A1? ->  
:TRIGGER:SOURCE:LOGIC:A1 RISE

**Description** This command is for setting or querying the edge pattern when the trigger is set to Edge on A, OR, or Window.

### **:TRIGger:TGTime?**

**Function** Queries all settings related to the T>Time trigger.

**Syntax** :TRIGger:TGTime?  
**Example** :TRIGGER:TGTIME? -> :TRIGGER:  
TGTIME:TIME 100.000E-09

### **:TRIGger:TGTime:TIME**

**Function** Sets the pulse width of the T>Time trigger or queries the current setting.

**Syntax** :TRIGger:TGTime:TIME {<Time>}  
:TRIGger:TGTime:TIME?  
<Time>=100 ns to 10 s  
**Example** :TRIGGER:TGTIME:TIME 600NS  
:TRIGGER:TGTIME:TIME? -> :TRIGGER:  
TGTIME:TIME 600.000E-09

### **:TRIGger:TIMER?**

**Function** Queries all settings related to the timer trigger.

**Syntax** :TRIGger:TIMER?  
**Example** :TRIGGER:TIMER? -> :TRIGGER:TIMER:  
DATE "99/01/01";TIME "00:00:00";  
INTERVAL HOUR1

### **:TRIGger:TIMER:DATE**

**Function** Sets the date of the timer trigger or queries the current setting.

**Syntax** :TRIGger:TIMER:DATE <String>  
:TRIGger:TIMER:DATE?  
<String>=YYYY/MM/DD  
**Example** :TRIGGER:TIMER:DATE "2002/06/04"  
:TRIGGER:TIMER:DATE? -> :TRIGGER:  
TIMER:DATE "2002/06/04"

### **:TRIGger:TImer:INTerval**

**Function** Sets the trigger time interval of the timer trigger or queries the current setting.

**Syntax** :TRIGger:TImer:INTerval {MIN1|MIN2|MIN3|MIN4|MIN5|MIN6|MIN7|MIN8|MIN9|MIN10|MIN15|MIN20|MIN25|MIN30|MIN40|MIN45|MIN50|HOUR1|HOUR2|HOUR3|HOUR4|HOUR5|HOUR6|HOUR7|HOUR8|HOUR9|HOUR10|HOUR11|HOUR12|HOUR18|HOUR24}  
:TRIGger:TImer:INTerval?  
**Example** :TRIGGER:TIMER:INTERVAL HOUR1  
:TRIGGER:TIMER:INTERVAL? ->  
:TRIGGER:TIMER:INTERVAL HOUR1

### **:TRIGger:TImer:TIME**

**Function** Sets the trigger time of the timer trigger or queries the current setting.

**Syntax** :TRIGger:TImer:TIME <String>  
:TRIGger:TImer:TIME?  
<String>=HH:MM:SS  
**Example** :TRIGGER:TIMER:TIME "12:34:56"  
:TRIGGER:TIMER:TIME? -> :TRIGGER:  
TIMER:TIME "12:34:56"

### **:TRIGger:TITime?**

**Function** Sets the pulse width of the TinTime trigger or queries the current setting.

**Syntax** :TRIGger:TITime?  
**Example** :TRIGGER:TITIME? -> :TRIGGER:  
TITIME:TIME1 100.000E-09;  
TIME2 200.000E-09

### **:TRIGger:TITime:TIME<x>**

**Function** Sets the pulse width of the TinTime trigger or queries the current setting.

**Syntax** :TRIGger:TITime:TIME<x> {<Time>}  
:TRIGger:TITime:TIME<x>?  
<Time>=100 ns to 10 s  
<x>=1 or 2  
**Example** :TRIGGER:TITIME:TIME1 100NS  
:TRIGGER:TITIME:TIME1? -> :TRIGGER:  
TITIME:TIME1 100.000E-09

### **:TRIGger:TLTime?**

**Function** Queries all settings related to the <Time trigger.

**Syntax** :TRIGger:TLTime?  
**Example** :TRIGGER:TLTIME? -> :TRIGGER:  
TLTIME:TIME 600.000E-09

### **:TRIGger:TLTime:TIME**

**Function** Sets the pulse width of the T<Time trigger or queries the current setting.

**Syntax** :TRIGger:TLTime:TIME {<Time>}  
:TRIGger:TLTime:TIME?  
<Time>=100 ns to 10 s  
**Example** :TRIGGER:TLTIME:TIME 500NS  
:TRIGGER:TLTIME:TIME? -> :TRIGGER:  
TLTIME:TIME 500.000E-09

### **:TRIGger:TOTime?**

**Function** Queries all settings related to the ToutTime trigger.

**Syntax** :TRIGger:TOTime?  
**Example** :TRIGGER:TOTIME? -> :TRIGGER:  
TOTIME:TIME1 200.000E-09;  
TIME2 300.000E-09

**:TRIGger:TOTime:TIME<x>**

Function Sets the pulse width of the ToutTime trigger or queries the current setting.

Syntax :TRIGger:TOTime:TIME<x> {<Time>}  
:TRIGger:TOTime:TIME<x>?  
<Time>=100 ns to 10 s  
<x>=1 or 2

Example :TRIGGER:TOTIME:TIME1 200NS  
:TRIGGER:TOTIME:TIME1? -> :TRIGGER:  
TOTIME:TIME1 200.000E-09

**:TRIGger:TYPE**

Function Sets the trigger type or queries the current setting.

Syntax :TRIGger:TYPE {ABN|ADB|EOA|BLTime|  
BGTime|BTOut|WINDOW|OR|SIMPlE|TGTime|  
TLTime|TITime|TOTime}  
:TRIGger:TYPE?

Example :TRIGGER:TYPE SIMPLE  
:TRIGGER:TYPE? -> :TRIGGER:  
TYPE SIMPLE

**:TRIGger:WINDOW?**

Function Queries all settings related to the window trigger.

Syntax :TRIGger:WINDOW?

Example :TRIGGER:WINDOW? -> :TRIGGER:  
WINDOW:CHANNEL1 IN;CHANNEL2 OFF;  
CHANNEL3 OFF;CHANNEL4 OFF;  
CHANNEL5 OFF;CHANNEL6 OFF;  
CHANNEL7 OFF;CHANNEL8 OFF;  
CHANNEL9 OFF;CHANNEL10 OFF;  
CHANNEL11 OFF;CHANNEL12 OFF;  
CHANNEL13 OFF;CHANNEL14 OFF;  
CHANNEL15 OFF;CHANNEL16 OFF;  
PODA DISABLE;PODB DISABLE

**:TRIGger:WINDOW:CHANnel<x>**

Function Sets the channel condition of the window trigger or queries the current setting.

Syntax :TRIGger:WINDOW:CHANnel<x>  
{IN|OUT|OFF}  
:TRIGger:WINDOW:CHANnel<x>?  
<x>=1 to 16

Example (The following is an example for CH1.)  
:TRIGGER:WINDOW:CHANNEL1 OUT  
:TRIGGER:WINDOW:CHANNEL1? ->  
:TRIGGER:WINDOW:CHANNEL1 OUT

**:TRIGger:WINDOW:{PODA|PODB}**

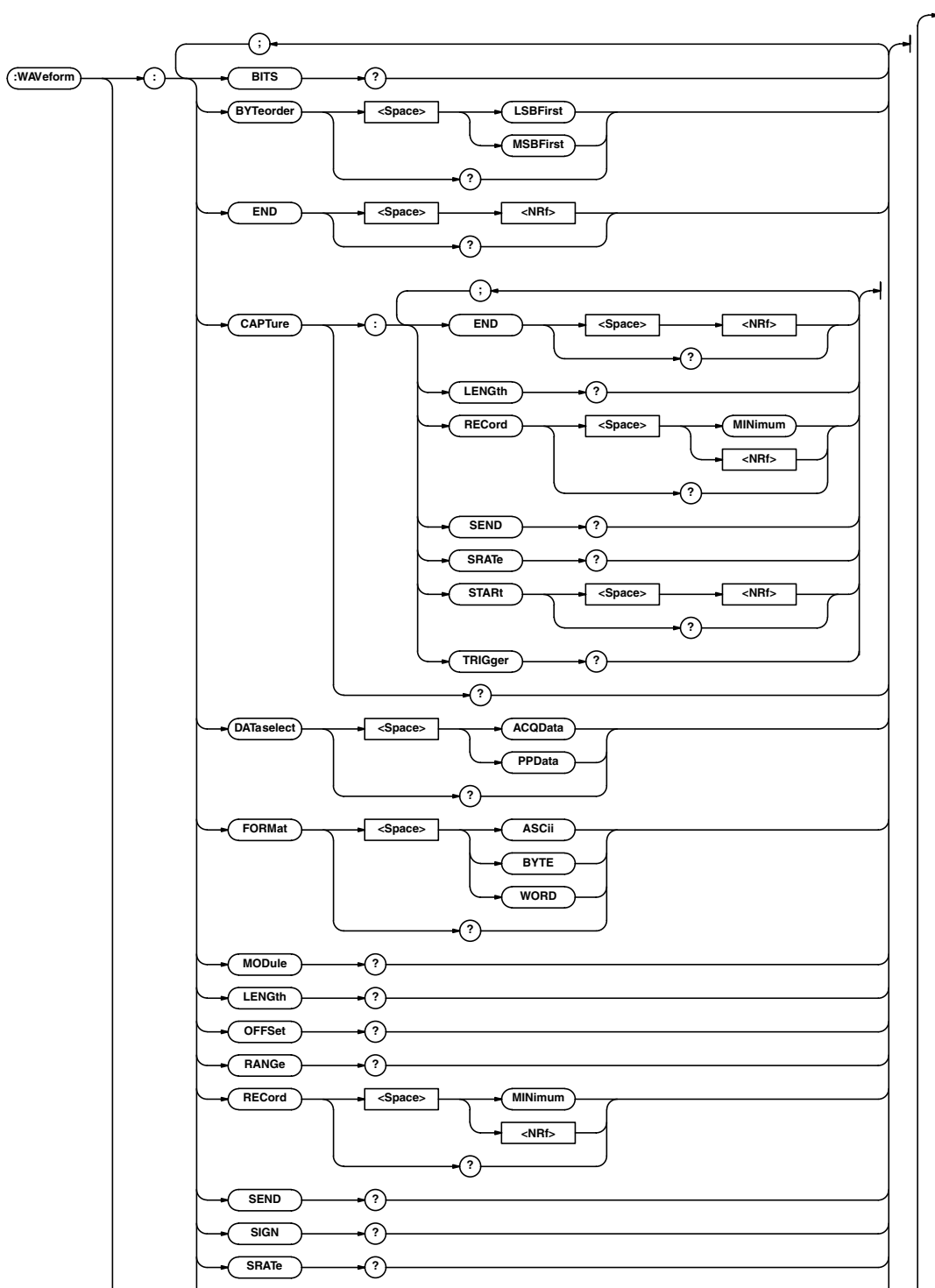
Function Sets the logic condition of the window trigger or queries the current setting.

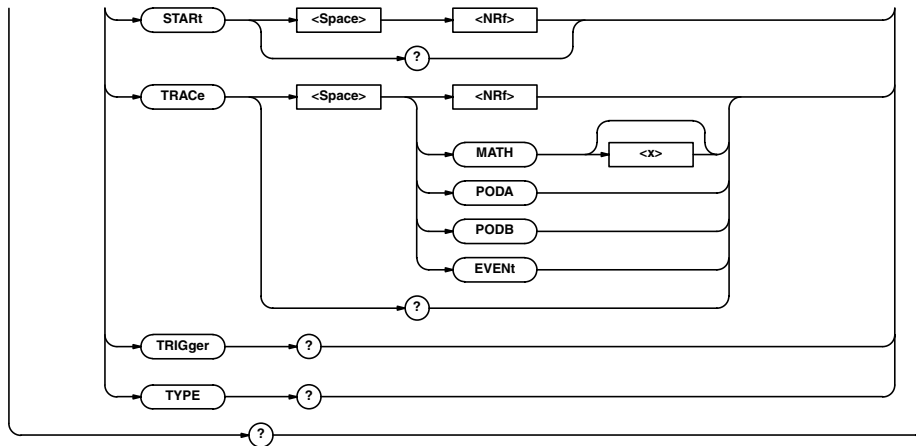
Syntax :TRIGger:WINDOW:{PODA|PODB}  
{DISable|ENABle}  
:TRIGger:WINDOW:{PODA|PODB}?

Example :TRIGGER:WINDOW:PODA ENABLE  
:TRIGGER:WINDOW:PODA? -> :TRIGGER:  
WINDOW:PODA ENABLE

### 6.31 WAVEform Group

The commands in this group deal with the acquired waveform data. There are no front panel keys that correspond to the commands in this group.



**:WAVEform?**

Function Queries all information about the waveform data.

Syntax :WAVEform?

Example :WAVEFORM? -> :WAVEFORM:TRACE 2;  
RECORD 0;START 0;END 0;  
DATASELECT ACQDATA;CAPTURE:  
RECORD 0;START 0;END 0;:WAVEFORM:  
FORMAT WORD;BYTEORDER LSBFIRST;  
MODULE M701250

**:WAVEform:BITS?**

Function Queries the bit length of the waveform data specified by ":WAVEform:TRACe".

Syntax :WAVEform:BITS?

Example :WAVEFORM:BITS? -> :WAVEFORM:  
BITS 16

**:WAVEform:BYTeorder**

Function Sets the transmission order when using word format of two bytes or more or queries the current setting.

Syntax :WAVEform:BYTeorder

{LSBFirst|MSBFirst}

:WAVEform:BYTeorder?

Example :WAVEFORM:BYTEORDER LSBFIRST  
:WAVEFORM:BYTEORDER? -> :WAVEFORM:  
BYTEORDER LSBFIRST

**:WAVEform:CAPTURE?**

Function Queries all information about the sub waveform.

Syntax :WAVEform:CAPTURE?

Example :WAVEFORM:CAPTURE? -> :WAVEFORM:  
CAPTURE:TRACE 2;RECORD 0;START 0;  
END 0

**:WAVEform:CAPTURE:END**

Function Sets the last data point of the sub waveform specified by :WAVEform:TRACe or queries the current setting.

Syntax :WAVEform:CAPTURE:END {<NRF>}

:WAVEform:CAPTURE:END?

<NRF>=0 to (total number of data points - 1)

Example :WAVEFORM:CAPTURE:END 0  
:WAVEFORM:CAPTURE:END? ->  
:WAVEFORM:CAPTURE:END 0

Description The total number of data points can be queried using :WAVEform:CAPTURE:LENGTH?.

**:WAVEform:CAPTURE:LENGTH?**

Function Queries the total number of points of the sub waveform specified by :WAVEform:TRACe.

Syntax :WAVEform:CAPTURE:LENGTH?

Example :WAVEFORM:CAPTURE:LENGTH? ->  
:WAVEFORM:CAPTURE:LENGTH 10010

**:WAVEform:CAPTURE:RECORD**

Function Sets the target record number of the sub waveform for the commands in the Waveform group or queries the current setting.

Syntax :WAVEform:CAPTURE:RECORD

{MAXimum|MINimum|<NRF>}

:WAVEform:CAPTURE:RECORD?

<NRF>=100 to -1999

Example :WAVEFORM:CAPTURE:RECORD 0  
:WAVEFORM:CAPTURE:RECORD? ->  
:WAVEFORM:CAPTURE:RECORD 0

Description Specifying "MAXimum" sets the record to the maximum record number. Specifying "MINimum" sets the record to the minimum record number.

**:WAVEform:CAPTURE:RECORD? MAXimum**

Function Queries the maximum record number of the sub waveform.

Syntax :WAVEform:CAPTURE:RECORD? MAXimum

Example :WAVEFORM:CAPTURE:RECORD? MAXimum  
-> :WAVEFORM:CAPTURE:RECORD 0

**:WAVEform:CAPTURE:RECORD? MINimum**

Function Queries the minimum record number of the sub waveform.

Syntax :WAVEform:CAPTURE:RECORD? MINimum

Example :WAVEFORM:CAPTURE:RECORD? MINimum  
-> :WAVEFORM:CAPTURE:RECORD -10

**:WAVEform:CAPTURE:SEND?**

Function Queries the sub waveform data specified by ":WAVEform:TRACe".

Syntax :WAVEform:CAPTURE:SEND?

Example :WAVEFORM:CAPTURE:SEND? -> #9  
#9(number of bytes, 9 digits) (data byte  
sequence)

Or, <NRF>,<NRF>,...

Description The output format of :WAVEform:CAPTURE:SEND? varies depending on the :WAVEform:FORMAT setting. For details, see :WAVEform:SEND?.

## 6.31 WAVEform Group

### **:WAVEform:CAPTURE:SRATE?**

**Function** Queries the sample rate of the record specified by :WAVEform:CAPTURE:RECORD.  
**Syntax** :WAVEform:CAPTURE:SRATE?  
**Example** :WAVEFORM:CAPTURE:SRATE? ->  
:WAVEFORM:CAPTURE:SRATE 10.0E+06

### **:WAVEform:CAPTURE:START**

**Function** Sets the first data point of the sub waveform specified by :WAVEform:TRACE or queries the current setting.  
**Syntax** :WAVEform:CAPTURE:START {<NRF>}  
:WAVEform:CAPTURE:START?  
{<NRF>}=0 to (total number of data points – 1)  
**Example** :WAVEFORM:CAPTURE:START 0  
:WAVEFORM:CAPTURE:START? ->  
:WAVEFORM:CAPTURE:START 0  
**Description** The total number of data points can be queried using :WAVEform:CAPTURE:LENGTH?.

### **:WAVEform:CAPTURE:TRIGGER?**

**Function** Queries the trigger position of the record specified by :WAVEform:CAPTURE:RECORD.  
**Syntax** :WAVEform:CAPTURE:TRIGGER?  
**Example** :WAVEFORM:CAPTURE:TRIGGER? ->  
:WAVEFORM:CAPTURE:TRIGGER 5000  
**Description** Queries the number of points from the first point of the record length to the trigger position.

### **:WAVEform:DATASElect**

**Function** Sets whether to query the waveform specified by :WAVEform:TRACE using ACQ data or P-P data.  
**Syntax** :WAVEform:DATASElect  
{ACQData|PPData}  
:WAVEform:DATASElect?  
**Example** :WAVEFORM:DATASELECT ACQDATA  
:WAVEFORM:DATASELECT? -> :WAVEFORM:  
DATASELECT ACQDATA

### **:WAVEform:END**

**Function** Sets the last data point of the main waveform specified by :WAVEform:TRACE or queries the current setting.  
**Syntax** :WAVEform:END {<NRF>}  
:WAVEform:END?  
<NRF>=0 to (total number of data points – 1)  
**Example** :WAVEFORM:END 0  
:WAVEFORM:END? -> :WAVEFORM:END 0  
**Description** The total number of data points can be queried using :WAVEform:LENGTH?.

### **:WAVEform:FORMat**

**Function** Sets the format of the data to be transmitted or queries the current setting.  
**Syntax** :WAVEform:FORMat {ASCIi|BYTE|WORD}  
:WAVEform:FORMat?  
**Example** :WAVEFORM:FORMAT WORD  
:WAVEFORM:FORMAT? -> :WAVEFORM:  
FORMAT WORD  
**Description** For details on the differences in the format setting, see the description of :WAVEform:SEND?.

### **:WAVEform:LENGth?**

**Function** Queries the total number of points of the main waveform specified by :WAVEform:TRACE.  
**Syntax** :WAVEform:LENGth?  
**Example** :WAVEFORM:LENGTH? -> :WAVEFORM:  
LENGTH 10020  
**Description** The total number of data points varies depending on the DL750 setting. For details, see appendix 1, “Relationship between the Time Axis Setting, Sample Rate and Record Length” in the *DL750 User's Manual*.

### **:WAVEform:MODUle?**

**Function** Queries the module corresponding to the waveform specified by “:WAVEform:TRACE”.  
**Syntax** :WAVEform:MODUle?  
**Example** :WAVEFORM:MODULE? -> :WAVEFORM:  
MODULE M701250  
**Description** The following values are returned from the module.

NOMODULE	No module
M701250	701250 (HS10M12)
M701251	701251 (HS1M16)
M701265	701265 (TEMP/HPV)

### **:WAVEform:OFFSet?**

**Function** Queries the offset value used when converting the waveform data specified by :WAVEform:TRACE to physical values.  
**Syntax** :WAVEform:OFFSet?  
**Example** :WAVEFORM:OFFSET? -> :WAVEFORM:  
OFFSET 0.0000000E+00  
**Description** This value is used when converting the <Block data> that is output using :WAVEform:SEND? to physical values.

### **:WAVEform:RANGe?**

**Function** Queries the range value when converting the waveform data specified by :WAVEform:TRACE to physical values.  
**Syntax** :WAVEform:RANGe?  
**Example** :WAVEFORM:RANGE? -> :WAVEFORM:  
RANGE 50.0000000E+00  
**Description** This value is used when converting the <Block data> that is output using :WAVEform:SEND? to physical values.

### **:WAVEform:RECORD**

**Function** Sets the target record number of the main waveform for the commands in the Waveform group or queries the current setting.  
**Syntax** :WAVEform:RECORD {MINimum|<NRF>}  
:WAVEform:RECORD?  
<NRF>=0 to –1999  
**Example** :WAVEFORM:RECORD 0  
:WAVEFORM:RECORD? -> :WAVEFORM:  
RECORD 0  
**Description** Specifying “MINimum” sets the record to the minimum record number. The selectable record number varies depending on the memory extension and acquisition setting. For details, see the *DL750 User's Manual*.

**:WAVEform:RECOrd? MINimum**

Function Queries the minimum record number of the history of the target channel.

Syntax :WAVEform:RECOrd? MINimum

Example :WAVEFORM:RECORD? MINIMUM ->  
:WAVEFORM:RECORD -10

**:WAVEform:SEND?**

Function Queries the main waveform data specified by “:WAVEform:TRACe”.

Syntax :WAVEform:SEND?

Example :WAVEFORM:SEND? -> #9  
(number of bytes, 9 digits)(data byte sequence)  
Or, <NRf>,<NRf>,...

Description The output format of :WAVEform:SEND? varies depending on the :WAVEform:FORMat setting.

(1) When set to ASCII

For the voltage module, the data is output as <Voltage>,<Voltage>,...,<Voltage>.

When measuring temperature, the data is output as

<Temperature>,<Temperature>,...,<Temperature>.

The unit specified by

:CHANnel<x>:TEMPerature:UNIT is used for the output.

For logic waveforms, the data is output as <NR1>,<NR1>,...,<NR1>.

<NR1> is 8-bit bit pattern (LSB: bit 1, MSB: bit 8) expressed in decimal notation.

For computed waveforms, the data is output as <NRf>,<NRf>,...,<NRf>.

(2) When set to BYTE or WORD

The data is output using the <Block data> format on all modules.

The data consists of signed values except for the logic module.

For voltage modules, the data can be converted to voltage using the following equation.

$$\text{Voltage} = \frac{\text{Range} \times \text{data} \times 10}{\text{Division}} + \text{Offset}$$

BYTE: Division=93.75

WORD: Division=24000

Range=Returned value of

:WAVEform:RANGE?

Offset=Returned value of

:WAVEform:OFFSet?

When measuring temperature, the data can be converted to temperature using the following equation.

$$\text{Temperature (}^\circ\text{C)} = \text{Data} \times \text{Division}$$

BYTE: Division=25.6

WORD: Division=0.1

For logic waveforms, the following format is used.

BYTE: 8-bit bit pattern (LSB: bit1, MSB: bit 8)

WORD: 16-bit bit pattern (upper 8 bits are PODB, lower 8 bits are the same as the PODA bit pattern)

For computed waveforms, the data can be converted using the following equation.

$$\text{Computed value} = \frac{\text{Range} \times \text{data} \times 10}{\text{Division}} + \text{Offset}$$

BYTE: Division=93.75

WORD: Division=24000

Range=Returned value of

:WAVEform:RANGE?

Offset=Returned value of

:WAVEform:OFFSet?

**:WAVEform:SIGN?**

Function Queries the existence of a sign when querying the waveform data specified by

:WAVEform:TRACe using block data.

Syntax :WAVEform:SIGN?

Example :WAVEFORM:SIGN? -> :WAVEFORM:SIGN 1

Description When the target waveform is a logic module waveform, 0 is returned. For all other waveforms, 1 is returned.

**:WAVEform:SRATE? (Sample RATE)**

Function Queries the sample rate of the record specified by :WAVEform:RECOrd.

Syntax :WAVEform:SRATE?

Example :WAVEFORM:SRATE? -> :WAVEFORM:  
SRATE 5.0E+03

**:WAVEform:START**

Function Sets the first data point of the main waveform specified by :WAVEform:TRACe or queries the current setting.

Syntax :WAVEform:START {<NRf>}  
:WAVEform:START?

{<NRf>}=0 to (total number of data points – 1)

Example :WAVEFORM:START 0  
:WAVEFORM:START? -> :WAVEFORM:  
START 0

Description The total number of data points can be queried using :WAVEform:LENGth?.

**:WAVEform:TRACe**

Function Sets the target waveform or queries the current setting.

Syntax :WAVEform:TRACe  
{<NRf> | MATH [<x>] | PODA | PODB | EVENT }

:WAVEform:TRACe?

{<NRf>}=1 to 16

<x>=1 to 8

Example :WAVEFORM:TRACe 1  
:WAVEFORM:TRACe? -> :WAVEFORM:  
TRACe 1

Description An error occurs, if a module is not installed in the channel (slot).

**:WAVEform:TRIGger?**

Function Queries the trigger position of the record specified by :WAVEform:RECOrd.

Syntax :WAVEform:TRIGger?

Example :WAVEFORM:TRIGGER? -> :WAVEFORM:  
TRIGGER 50005

Description Queries the number of points from the first point of the record to the trigger position.



## 6.31 WAVEform Group/6.32 XY Group

### :WAVEform:TYPE?

**Function** Queries the acquisition mode of the waveform specified by :WAVEform:TRACE.

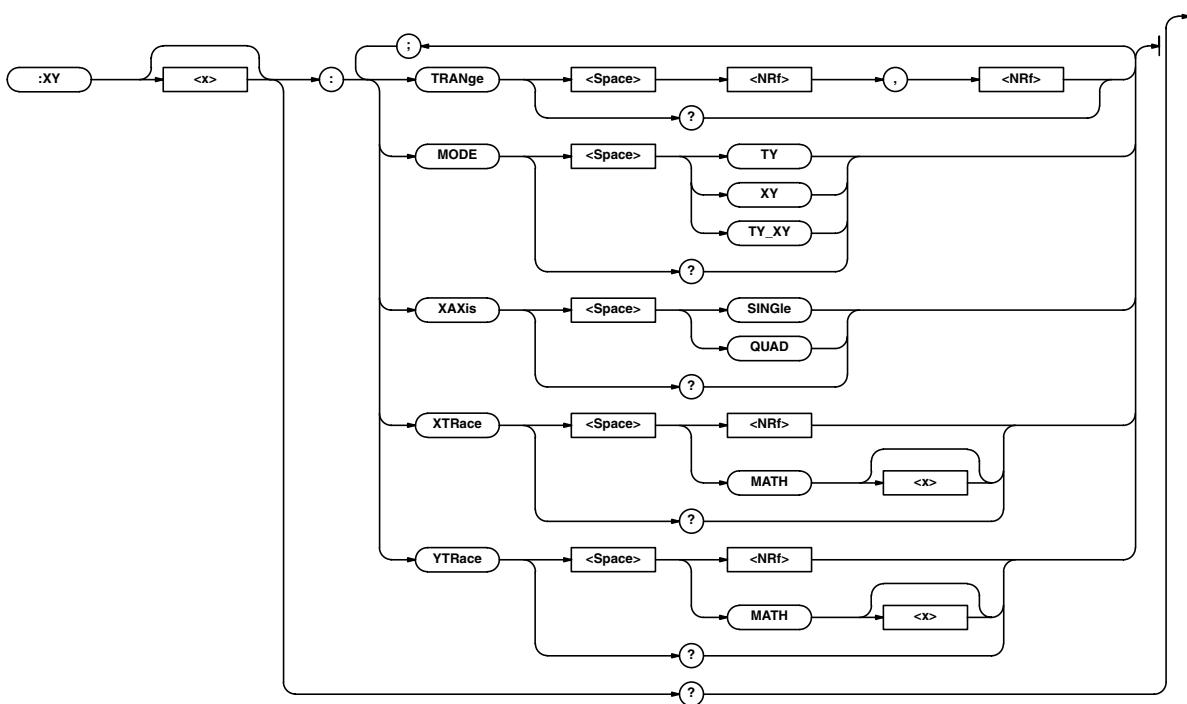
**Syntax** :WAVEform:TYPE?

**Example** :WAVEFORM:TYPE? -> :WAVEFORM:TYPE NORMAL

**Description** AVERage, ENVelope, BAVerage, or NORMal is returned.

## 6.32 XY Group

The commands in this group deal with the XY display. You can make the same settings and inquiries as when the X-Y key (SHIFT+DISPLAY) on the front panel is used.



### :XY<x>?

**Function** Queries all settings related to the X-Y display.

**Syntax** :XY<x>?  
<x>=1 to 4

**Example** :XY1? -> :XY1:MODE XY;XAXIS SINGLE;  
DISPLAY 1;TRANGE -5.00,5.00;  
XTRACE 1;YTRACE 2

### :XY<x>:MODE

**Function** Sets the display mode or queries the current setting.

**Syntax** :XY<x>:MODE {TY|TY\_XY|XY}  
:XY<x>:MODE?  
<x>=1 to 4

**Example** :XY1:MODE XY  
:XY1:MODE? -> :XY1:MODE XY

**Description** The same setting or query is performed regardless of the <x> value (any value from 1 to 4).

### :XY<x>:TRAnge

**Function** Sets the range of the T-Y waveform to be displayed on the X-Y display or queries the current setting.

**Syntax** :XY<x>:TRAnge {<NRf>,<NRf>}  
:XY<x>:TRAnge?  
<x>=1 to 4  
<NRf>=-5 to 5 divisions (10 divisions/displayed record length steps)

**Example** :XY1:TRANGE -4,4  
:XY1:TRANGE? -> :XY1:TRANGE -4.00,4.00

**Description** The same setting or query is performed regardless of the <x> value (any value from 1 to 4).

**:XY<x>:XAXis**

**Function** Sets whether to make the channel assigned to the X-axis of the X-Y display common (SINGLE) or separate (QUAD) or queries the current setting.

**Syntax** :XY<x>:XAXis {SINGLE|QUAD}  
:XY<x>:XAXis?  
<x>=1 to 4

**Example** :XY1:XAXIS SINGLE  
:XY1:XAXIS? -> :XY1:XAXIS SINGLE

**Description** The same setting or query is performed regardless of the <x> value (any value from 1 to 4).

**:XY<x>:XTRace**

**Function** Sets the channel to assign to the X-axis of the X-Y display or queries the current setting.

**Syntax** :XY<x>:XTRace {<NRf>|MATH<x>}  
:XY<x>:XTRace?  
<x> of XY<x>=1 to 4  
<NRf>=1 to 16  
<x> of MATH<x>=1 to 8

**Example** :XY1:XTRACE 1  
:XY1:XTRACE? -> :XY1:XTRACE 1

**Description** There is a limitation on the channels that can be specified when :XY<x>:XAXis is set to QUAD.

**:XY<x>:YTRace**

**Function** Sets the channel to assign to the Y-axis of the X-Y display or queries the current setting.

**Syntax** :XY<x>:YTRace {<NRf>|MATH<x>}  
:XY<x>:YTRace?  
<x> of XY<x>=1 to 4  
<NRf>=1 to 16  
<x> of MATH<x>=1 to 8

**Example** :XY1:YTRACE 1  
:XY1:YTRACE? -> :XY1:YTRACE 1

**Description** This command is valid when :XY<x>:XAXis is set to QUAD. There is a limitation on the channels that can be specified.

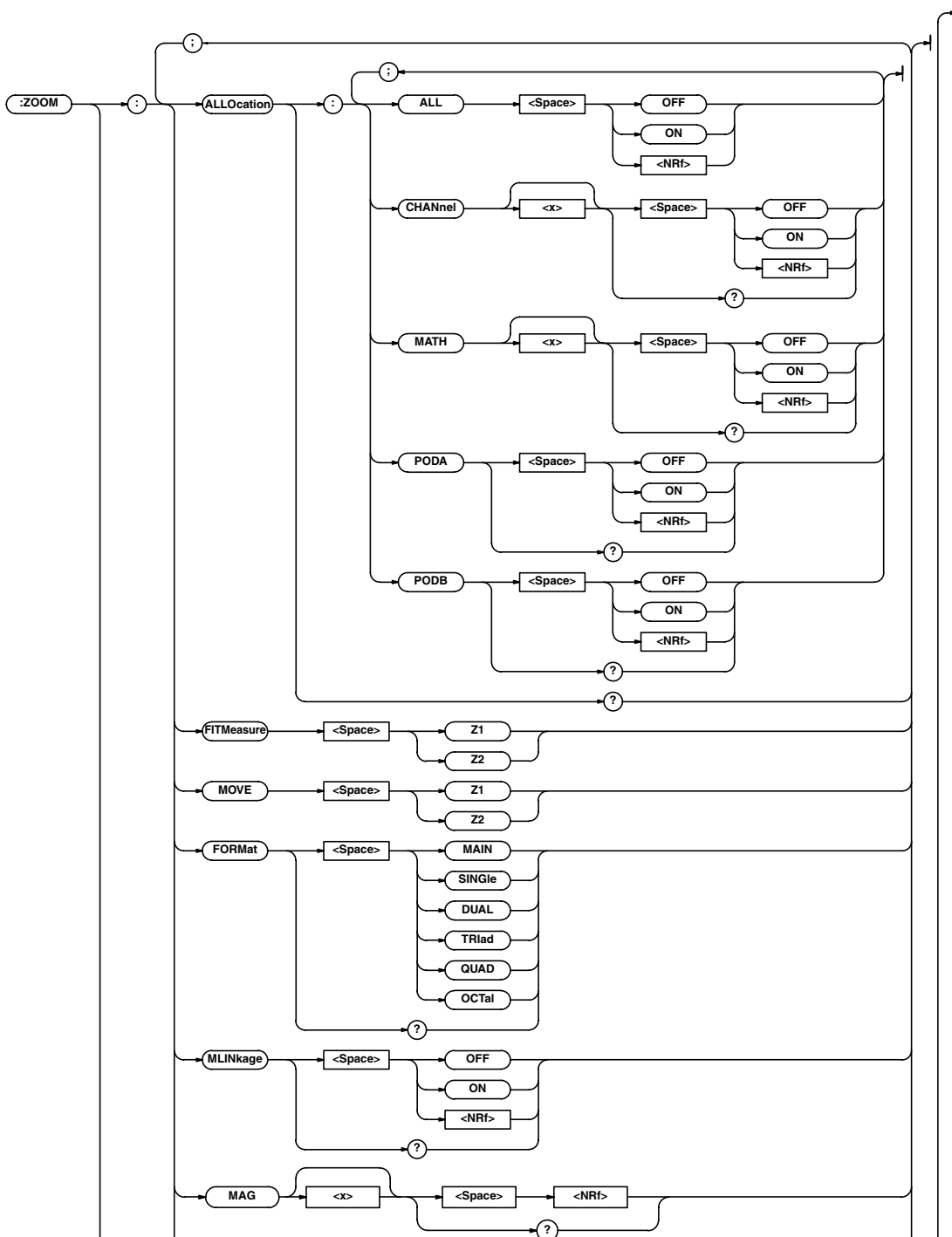
## 6.33 ZOOM Group

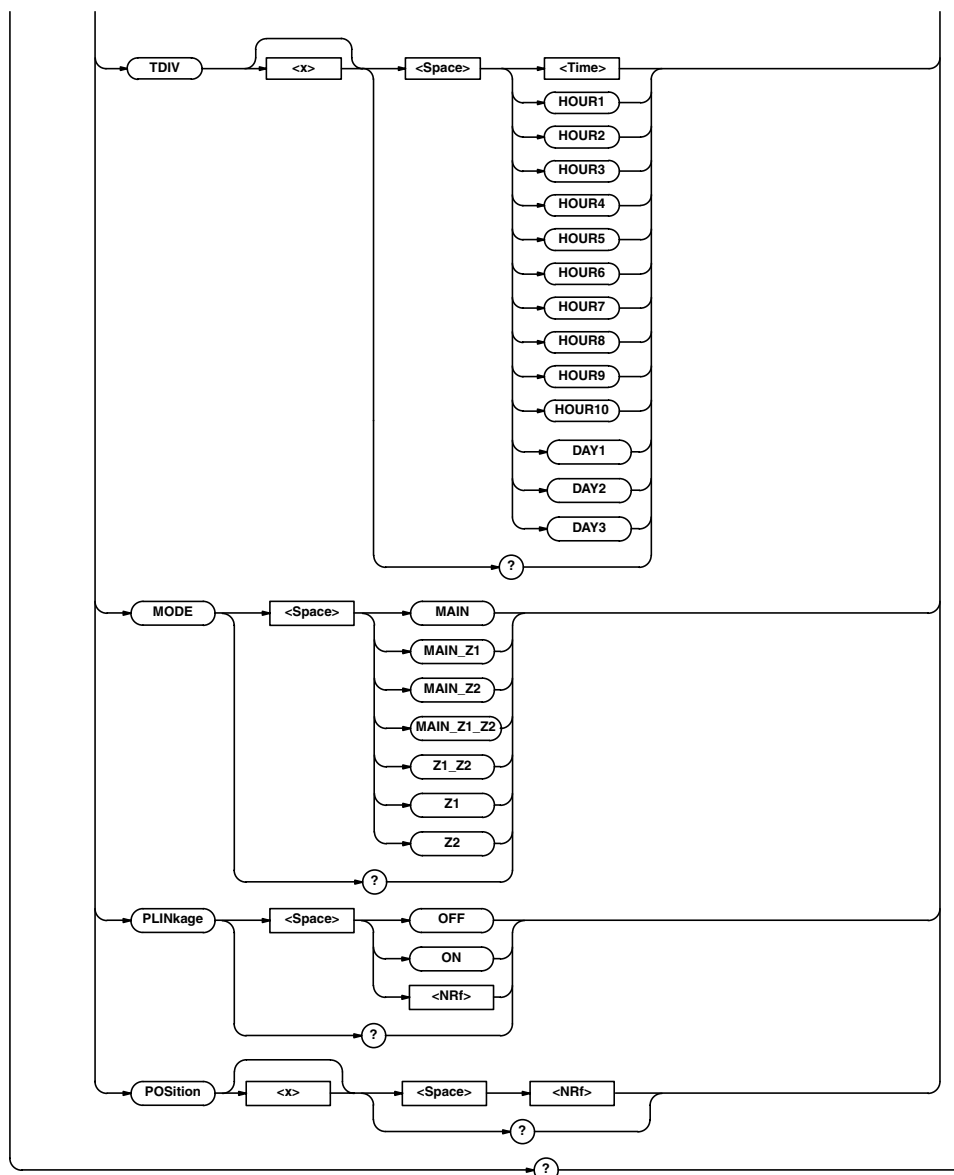
### 6.33 ZOOM Group

The commands in this group deal with the ZOOM display. You can make the same settings and inquiries as when ZOOM on the front panel is used.

When the time axis range setting is as follows, the selectable range varies depending on the record length setting. For details, see "Selectable Range of Cursor Position" in section 8.1 of the DL750 User's Manual.

<NRf>=-5 to 5 div



**:ZOOM?**

Function Queries all settings related to the waveform zoom.

Syntax :ZOOM?

Example :ZOOM? -> :ZOOM:MODE MAIN;  
 FORMAT MAIN;ALLOCATION:CHANNEL1 1;  
 CHANNEL2 1;CHANNEL3 1;CHANNEL4 1;  
 CHANNEL5 1;CHANNEL6 1;CHANNEL7 1;  
 CHANNEL8 1;CHANNEL9 1;CHANNEL10 1;  
 CHANNEL11 1;CHANNEL12 1;  
 CHANNEL13 1;CHANNEL14 1;  
 CHANNEL15 1;CHANNEL16 1;MATH1 0;  
 MATH2 0;MATH3 0;MATH4 0;MATH5 0;  
 MATH6 0;MATH7 0;MATH8 0;PODA 0;  
 PODB 0;;ZOOM:MLINKAGE 1;  
 MAG1 500.0E-06;MAG2 500.0E-06;  
 PLINKAGE 0;POSITION1 0.0000000;  
 POSITION2 0.0000000

**:ZOOM:ALlocation?**

Function Queries all settings related to the zoom target waveform.

Syntax :ZOOM:ALlocation?

Example :ZOOM:ALLOCATION? -> :ZOOM:  
 ALLOCATION:CHANNEL1 1;CHANNEL2 1;  
 CHANNEL3 1;CHANNEL4 1;CHANNEL5 1;  
 CHANNEL6 1;CHANNEL7 1;CHANNEL8 1;  
 CHANNEL9 1;CHANNEL10 1;CHANNEL11 1;  
 CHANNEL12 1;CHANNEL13 1;  
 CHANNEL14 1;CHANNEL15 1;  
 CHANNEL16 1;MATH1 0;MATH2 0;  
 MATH3 0;MATH4 0;MATH5 0;MATH6 0;  
 MATH7 0;MATH8 0;PODA 0;PODB 0

**:ZOOM:ALlocation:ALL**

Function Sets all waveforms to be zoomed.

Syntax :ZOOM:ALlocation:ALL {<Boolean>}

Example :ZOOM:ALLOCATION:ALL ON

## 6.33 ZOOM Group

### **:ZOOM:ALlocation**

**:{CHANnel<x>|MATH<x>|PODA|PODB}**

**Function** Sets the zoom target waveform or queries the current setting.

**Syntax** :ZOOM:ALlocation:{CHANnel<x>|  
MATH<x>|PODA|PODB} {<Boolean>}  
:ZOOM:ALlocation:{CHANnel<x>|  
MATH<x>|PODA|PODB}?  
<x> of CHANnel<x>=1 to 16  
<x> of MATH<x>=1 to 8

**Example** :ZOOM:ALLOCATION:CHANNEL1 ON  
:ZOOM:ALLOCATION:CHANNEL1? ->  
:ZOOM:ALLOCATION:CHANNEL1 1

**Description** An error occurs, if a module is not installed in the channel (slot).

### **:ZOOM:FITMeasure**

**Function** Moves the range of the automated measurement of waveform parameters to the ends of the zoom box.

**Syntax** :ZOOM:FITMeasure {Z1|Z2}  
**Example** :ZOOM:FITMEASURE Z1

### **:ZOOM:MOVE**

**Function** Moves the zoom box to the beginning of the waveform.

**Syntax** :ZOOM:MOVE {Z1|Z2}  
**Example** :ZOOM:MOVE Z1

### **:ZOOM:FORMat**

**Function** Sets the zoom waveform display format or queries the current setting.

**Syntax** :ZOOM:FORMat  
{MAIN|SINGle|DUAL|TRIad|QUAD|OCTal}  
:ZOOM:FORMat?  
**Example** :ZOOM:FORMAT SINGLE  
:ZOOM:FORMAT? -> :ZOOM:  
FORMAT SINGLE

### **:ZOOM:MLINKage (Mag Linkage)**

**Function** Turns ON/OFF the link to the T/div value of the zoom waveform or queries the current setting.

**Syntax** :ZOOM:MLINKage {<Boolean>}  
:ZOOM:MLINKage?

**Example** :ZOOM:MLINKAGE ON  
:ZOOM:MLINKAGE? -> :ZOOM:MLINKAGE 1

### **:ZOOM:MAG<x>**

**Function** Sets the zoom rate of the zoom window or queries the current setting.

**Syntax** :ZOOM:TDIV<x> {<NRf>}  
:ZOOM:TDIV<x>?  
<x>=1 or 2  
<NRf>=2 to 100000000 (see the DL750 User's Manual)

**Example** :ZOOM:MAG1 10  
:ZOOM:MAG1? -> :ZOOM:MAG1 10.0

**Description** When zooming a waveform measured using the external clock, set the magnification.

### **:ZOOM:MODE**

**Function** Sets the zoom waveform display format or queries the current setting.

**Syntax** :ZOOM:MODE {MAIN|MAIN\_Z1|MAIN\_Z2|  
MAIN\_Z1\_Z2|Z1\_Z2|Z1|Z2}  
:ZOOM:MODE?

**Example** :ZOOM:MODE MAIN\_Z1\_Z2  
:ZOOM:MODE? -> :ZOOM:  
MODE MAIN\_Z1\_Z2

### **:ZOOM:PLINKage (Position Linkage)**

**Function** Turns ON/OFF the link when moving the zone box or queries the current setting.

**Syntax** :ZOOM:PLINKage {<Boolean>}  
:ZOOM:PLINKage?

**Example** :ZOOM:PLINKAGE ON  
:ZOOM:PLINKAGE? -> :ZOOM:PLINKAGE 1

### **:ZOOM:POSition<x>**

**Function** Sets the zoom box position or queries the current setting.

**Syntax** :ZOOM:POSition<x> {<NRf>}  
:ZOOM:POSition<x>?  
<x>=1 or 2

<NRf>=-5 to 5 div

**Example** :ZOOM:POSITION1 0  
:ZOOM:POSITION1? -> :ZOOM:  
POSITION1 0.0000000

### **:ZOOM:TDIV<x>**

**Function** Sets the T/div value of the zoom waveform or queries the current setting.

**Syntax** :ZOOM:TDIV<x> {<Time>|HOUR1|HOUR2|  
HOUR3|HOUR4|HOUR5|HOUR6|HOUR7|HOUR8|  
HOUR9|HOUR10|HOUR12|DAY1|DAY2|DAY3}  
:ZOOM:TDIV<x>?  
<x>=1 or 2

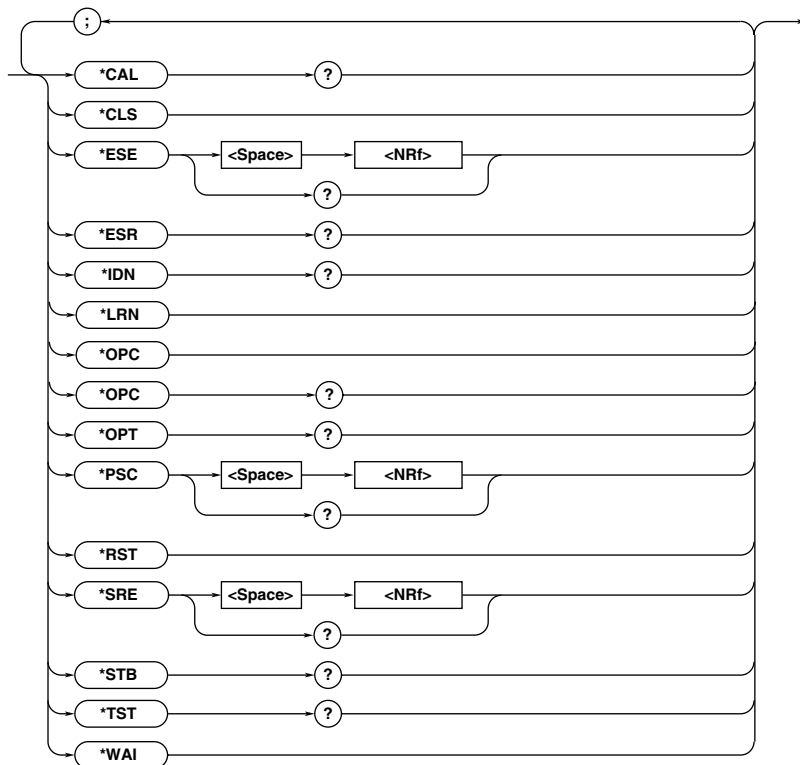
<Time>=See the *DL750 User's Manual*

**Example** :ZOOM:TDIV1 200MS  
:ZOOM:TDIV1? -> :ZOOM:TDIV1 200.0E-  
03

**Description** When zooming waveforms measured using the internal clock, the T/div value is set in terms of time.

## 6.34 Common Command Group

The commands in the common group are defined in the IEEE 488.2-1987 and are independent of the instrument's functions. There are no front panel keys that correspond to the commands in this group.



### \*CAL? (CALibrate)

**Function** Performs calibration and queries the result.  
**Syntax** \*CAL?  
**Example** \*CAL? -> 0  
**Description** If the calibration terminates normally, 0 is returned. If an error is detected, 1 is returned.

### \*CLS (CLear Status)

**Function** Clears the standard event register, extended event register, and error queue.  
**Syntax** \*CLS  
**Example** \*CLS  
**Description**

- If the \*CLS command is located immediately after the program message terminator, the output queue is also cleared.
- For details on the register and queue, see chapter 7.

### \*ESE (standard Event Status Enable register)

**Function** Sets the standard event enable register or queries the current setting.  
**Syntax** \*ESE {<NRf>}  
 \*ESE?  
 <NRf>=0 to 255  
**Example** \*ESE 251  
 \*ESE? -> 251  
**Description**

- Specify the value as a sum of decimal values of each bit.
- For example, specifying “\*ESE 251” will cause the standard enable register to be set to “11111011.” In this case, bit 2 of the standard event register is disabled which means that bit 5 (ESB) of the status byte register is not set to “1,” even if a “query error” occurs.
- The default value is “\*ESE 0” (all bits disabled).
- A query using \*ESE? will not clear the contents of the standard event enable register.
- For details on the standard event enable register, see page 7-3.

## 6.34 Common Command Group

### \*ESR? (standard Event Status Register)

**Function** Queries the standard event register and clears the register.

**Syntax** \*ESR?

**Example** \*ESR? -> 32

**Description**

- A sum of decimal values of each bit is returned.
- You can check what type of events occurred when an SRQ is generated.
- For example, if a value of "32" is returned, this indicates that the standard event register is set to "00100000." In this case, you can see that the SRQ occurred due to a "command syntax error."
- A query using \*ESR? will clear the contents of the standard event register.
- For details on the standard event register, see page 7-3.

### \*IDN? (IDeNtify)

**Function** Queries the instrument model.

**Syntax** \*IDN?

**Example** \*IDN? -> YOKOGAWA,701210,0,F1.10

**Description** The information is returned in the following form: <Manufacturer>,<Model>,<Serial No.>,<Firmware version>  
In actuality, <Serial No.> is not returned (always 0).

### \*LRN? (LeaRN)

**Function** Queries collectively the current settings of the following command groups.  
ACQuire, CHANnel<x>, TIMEbase, TRIGger

**Syntax** \*LRN?

**Example** \*LRN? -> :ACQUIRE:RLENGTH 10000;  
MODE NORMAL;PROTATE 1;AVERAGE:  
COUNT INFINITY;EWEIGHT 16;:  
ACQUIRE:COUNT INFINITY;  
CLOCK INTERNAL;RTOUT:MODE OFF;  
ANAMING DATE;FILENAME "";  
COMMENT "";:CHANNEL1:DISPLAY 1;  
VOLTAGE:VDIV 50.000E+00;VARIABLE 0;  
COUPLING DC;PROBE 10;BWIDTH FULL;  
SCALE 250.000E+00,-250.000E+00;  
POSITION 0.00;LSCALE:MODE OFF;  
AVALUE 1.00000E+00;  
BVALUE 0.00000E+00;P1X 0.00000E+00;  
P1Y 0.00000E+00;P2X 1.00000E+00;  
P2Y 1.00000E+00;UNIT "";:CHANNEL1:  
VOLTAGE:ZOOM 1.000;INVERT 0;  
OFFSET 0.0E+00;:CHANNEL1:  
TEMPERATURE:COUPLING  
DC;:CHANNEL1:LABEL "CH1";:CHANNEL2:  
DISPLAY 1;VOLTAGE:VDIV 50.000E+00;  
VARIABLE 0;COUPLING DC;PROBE 10;  
BWIDTH FULL;SCALE 250.000E+00,  
-250.000E+00;POSITION 0.00;  
LSCALE:MODE OFF;AVALUE 1.00000E+00;  
BVALUE 0.00000E+00;P1X 0.00000E+00;  
P1Y 0.00000E+00;P2X 1.00000E+00;  
P2Y 1.00000E+00;UNIT "";:CHANNEL2:  
VOLTAGE:ZOOM 1.000;INVERT 0;  
OFFSET 0.0E+00;:CHANNEL2:  
TEMPERATURE:COUPLING DC;:CHANNEL2:  
LABEL "CH2";:CHANNEL3:DISPLAY 1;  
VOLTAGE:VDIV 50.000E+00;VARIABLE 0;

COUPLING DC;PROBE 10;BWIDTH FULL;  
SCALE 250.000E+00,-250.000E+00;  
POSITION 0.00;LSCALE:MODE OFF;  
AVALUE 1.00000E+00;  
BVALUE 0.00000E+00;P1X 0.00000E+00;  
P1Y 0.00000E+00;P2X 1.00000E+00;  
P2Y 1.00000E+00;UNIT "";:CHANNEL3:  
VOLTAGE:ZOOM 1.000;INVERT 0;  
OFFSET 0.0E+00;:CHANNEL3:  
TEMPERATURE:COUPLING DC;:CHANNEL3:  
LABEL "CH3";:CHANNEL4:DISPLAY 1;  
VOLTAGE:VDIV 50.000E+00;VARIABLE 0;  
COUPLING DC;PROBE 10;BWIDTH FULL;  
SCALE 250.000E+00,-250.000E+00;  
POSITION 0.00;LSCALE:MODE OFF;  
AVALUE 1.00000E+00;  
BVALUE 0.00000E+00;P1X 0.00000E+00;  
P1Y 0.00000E+00;P2X 1.00000E+00;  
P2Y 1.00000E+00;UNIT "";:CHANNEL4:  
VOLTAGE:ZOOM 1.000;INVERT 0;  
OFFSET 0.0E+00;:CHANNEL4:  
TEMPERATURE:COUPLING DC;:CHANNEL4:  
LABEL "CH4";:CHANNEL5:DISPLAY 1;  
VOLTAGE:VDIV 50.000E+00;VARIABLE 0;  
COUPLING DC;PROBE 10;BWIDTH FULL;  
SCALE 250.000E+00,-250.000E+00;  
POSITION 0.00;LSCALE:MODE OFF;  
AVALUE 1.00000E+00;  
BVALUE 0.00000E+00;P1X 0.00000E+00;  
P1Y 0.00000E+00;P2X 1.00000E+00;  
P2Y 1.00000E+00;UNIT "";:CHANNEL5:  
VOLTAGE:ZOOM 1.000;INVERT 0;  
OFFSET 0.0E+00;:CHANNEL5:  
TEMPERATURE:COUPLING DC;:CHANNEL5:  
LABEL "CH5";:CHANNEL6:DISPLAY 1;  
VOLTAGE:VDIV 50.000E+00;VARIABLE 0;  
COUPLING DC;PROBE 10;BWIDTH FULL;  
SCALE 250.000E+00,-250.000E+00;  
POSITION 0.00;LSCALE:MODE OFF;  
AVALUE 1.00000E+00;  
BVALUE 0.00000E+00;P1X 0.00000E+00;  
P1Y 0.00000E+00;P2X 1.00000E+00;  
P2Y 1.00000E+00;UNIT "";:CHANNEL6:  
VOLTAGE:ZOOM 1.000;INVERT 0;  
OFFSET 0.0E+00;:CHANNEL6:  
TEMPERATURE:COUPLING DC;:CHANNEL6:  
LABEL "CH6";:EVENT:DISPLAY 0;  
POSITION 0.00;ZOOM 1.000;VOICE 0;  
DCAPTURE 1;LABEL "Event";:LOGIC:  
PODA:DISPLAY 0;POSITION 0.00;  
ZOOM 1.000;LABEL "LogicA";BIT1:  
DISPLAY 1;LABEL "A-1";:LOGIC:PODA:  
BIT2:DISPLAY 1;LABEL "A-2";:LOGIC:  
PODA:BIT3:DISPLAY 1;LABEL "A-3";:  
LOGIC:PODA:BIT4:DISPLAY 1;  
LABEL "A-4";:LOGIC:PODA:BIT5:  
DISPLAY 1;LABEL "A-5";:LOGIC:PODA:  
BIT6:DISPLAY 1;LABEL "A-6";:LOGIC:  
PODA:BIT7:DISPLAY 1;LABEL "A-7";:  
LOGIC:PODA:BIT8:DISPLAY 1  
;LABEL "A-8";:LOGIC:PODA:  
BMAPPING AUTO;:LOGIC:PODB:  
DISPLAY 0;POSITION 0.00;ZOOM 1.000;  
LABEL "LogicB";BIT1:DISPLAY 1;  
LABEL "B-1";:LOGIC:PODB:BIT2:  
DISPLAY 1;LABEL "B-2";:LOGIC:PODB:  
BIT3:DISPLAY 1;LABEL "B-3";:LOGIC:  
PODB:BIT4:DISPLAY 1;LABEL "B-4";:

```

LOGIC:PODB:BIT5:DISPLAY 1;
LABEL "B-5";:LOGIC:PODB:BIT6:
DISPLAY 1;LABEL "B-6";:LOGIC:PODB:
BIT7:DISPLAY 1;LABEL "B-7";:LOGIC:
PODB:BIT8:DISPLAY 1;LABEL "B-8";:
LOGIC:PODB:BMAPPING AUTO;:
TIMEBASE:SOURCE INTERNAL;
TDIV 100.0E-03;:TRIGGER:ACTION:
MODE 0;HCOPY 0;IMAGE 0;SAVE 0;
BUZZER 0;SEQUENCE CONTINUE;:
TRIGGER:SIMPLE:SOURCE 1;
LEVEL 0.0E+00;HYSTERESIS LOW;
SLOPE RISE;LOGIC:BIT 1;SLOPE RISE;:
TRIGGER:ABN:COUNT 1;:TRIGGER:ADB:
DELAY 0.0E+00;:TRIGGER:MODE AUTO;
EOA:CHANNEL1 OFF;CHANNEL2 RISE;
CHANNEL3 OFF;CHANNEL4 OFF;
CHANNEL5 OFF;CHANNEL6 OFF;
PODA DISABLE;PODB DISABLE;
CONDITION TRUE;:TRIGGER:OR:
CHANNEL1 RISE;CHANNEL2 OFF;
CHANNEL3 OFF;CHANNEL4 OFF;
CHANNEL5 OFF;CHANNEL6 OFF;
PODA DISABLE;PODB DISABLE;:TRIGGER:
TYPE SIMPLE;BGTIME:
TIME 100.000E-09;:TRIGGER:BLTIME:
TIME 100.000E-09;:TRIGGER:BTOUT:
TIME 100.000E-09;:TRIGGER:TGTIME:
TIME 100.000E-09;:TRIGGER:TLTIME:
TIME 100.000E-09;:TRIGGER:TITIME:
TIME1 100.000E-09;
TIME2 200.000E-09;:TRIGGER:TOTIME:
TIME1 100.000E-09;
TIME2 200.000E-09;:TRIGGER:TIMER:
DATE "2002/01/01";TIME "00:00:00";
INTERVAL HOUR1;:TRIGGER:WINDOW:
CHANNEL1 IN;CHANNEL2 OFF;
CHANNEL3 OFF;CHANNEL4 OFF;
CHANNEL5 OFF;CHANNEL6 OFF;
PODA DISABLE;PODB DISABLE;:TRIGGER:
HOLDOFF:TIME 0.0E+00;:TRIGGER:
SOURCE:CHANNEL1:LEVEL 0.0E+00;
HYSTERESIS LOW;CENTER 0.0E+00;
WIDTH 50.000E+00;:TRIGGER:SOURCE:
CHANNEL2:LEVEL 0.0E+00;
HYSTERESIS LOW;CENTER 0.0E+00;
WIDTH 50.000E+00;:TRIGGER:SOURCE:
CHANNEL3:LEVEL 0.0E+00;
HYSTERESIS LOW;CENTER 0.0E+00;
WIDTH 50.000E+00;:TRIGGER:SOURCE:
CHANNEL4:LEVEL 0.0E+00;HYSTERESIS
LOW;CENTER 0.0E+00;
WIDTH 50.000E+00;:TRIGGER:SOURCE:
CHANNEL5:LEVEL 0.0E+00;
HYSTERESIS LOW;CENTER 0.0E+00;
WIDTH 50.000E+00;:TRIGGER:SOURCE:
CHANNEL6:LEVEL 0.0E+00;
HYSTERESIS LOW;CENTER 0.0E+00;
WIDTH 50.000E+00;:TRIGGER:SOURCE:
LOGIC:A1 RISE;A2 DONTCARE;
A3 DONTCARE;A4 DONTCARE;
A5 DONTCARE;A6 DONTCARE;
A7 DONTCARE;A8 DONTCARE;
B1 DONTCARE;B2 DONTCARE;
B3 DONTCARE;B4 DONTCARE;
B5 DONTCARE;B6 DONTCARE;
B7 DONTCARE;B8 DONTCARE;:TRIGGER:
ATRIGGER:CHANNEL1 HIGH;

```

```

CHANNEL2 DONTCARE;
CHANNEL3 DONTCARE;
CHANNEL4 DONTCARE;
CHANNEL5 DONTCARE;
CHANNEL6 DONTCARE;PODA DISABLE;
PODB DISABLE;LOGIC:A1 DONTCARE;
A2 HIGH;A3 DONTCARE;A4 DONTCARE;
A5 DONTCARE;A6 DONTCARE;
A7 DONTCARE;A8 DONTCARE;
B1 DONTCARE;B2 DONTCARE;
B3 DONTCARE;B4 DONTCARE;
B5 DONTCARE;B6 DONTCARE;
B7 DONTCARE;B8 DONTCARE;:TRIGGER:
ATRIGGER:CONDITION ENTER;:TRIGGER:
BTRIGGER:CHANNEL1 DONTCARE;
CHANNEL2 HIGH;CHANNEL3 DONTCARE;
CHANNEL4 DONTCARE;
CHANNEL5 DONTCARE;
CHANNEL6 DONTCARE;PODA DISABLE;
PODB DISABLE;LOGIC:A1 HIGH;
A2 DONTCARE;A3 DONTCARE;
A4 DONTCARE;A5 DONTCARE;
A6 DONTCARE;A7 DONTCARE;
A8 DONTCARE;B1 DONTCARE;
B2 DONTCARE;B3 DONTCARE;
B4 DONTCARE;B5 DONTCARE;
B6 DONTCARE;B7 DONTCARE;
B8 DONTCARE;:TRIGGER:BTRIGGER:
CONDITION ENTER;:TRIGGER:
DELAY 0.0E+00;POSITION 50.000;
SCOUNT 1

```

**\*OPC (Operation Complete)**

**Function** Sets a "1" to bit 0 (OPC bit) of the standard event register bit upon the completion of the specified overlap command.

**Syntax** \*OPC

**Example** \*OPC

**Description**

- For the description regarding how to synchronize the program using \*OPC, see page 5-8.
- The :COMMunicate:OPSE command is used to specify the overlap command.
- If \*OPC is not the last command of the message, the operation is not guaranteed.

**\*OPC? (Operation Complete)**

**Function** If \*OPC? is transmitted and the specified overlap command is completed, ASCII code "1" is returned.

**Syntax** \*OPC?

**Example** \*OPC? -> 1

**Description**

- For the description regarding how to synchronize the program using \*OPC?, see page 5-8.
- The :COMMunicate:OPSE command is used to specify the overlap command.
- If \*OPC? is not the last command of the message, the operation is not guaranteed.



## 6.34 Common Command Group

### **\*OPT? (OPTion)**

Function Queries the installed options.  
Syntax \*OPT?  
Example \*OPT? -> CH25MW, PCCARD, HD, ETHER, USERDEFINE, PROBEPOWER  
Description

- Returns the presence of <Channel>, <HD>, <User-defined equation>, and <32-bit extended logic input>
- The \*OPT? query must be the last query of the program message. An error occurs if there is a query after this query.

### **\*PSC (Power-on Status Clear)**

Function Sets whether or not to clear the registers below at power on or queries the current setting. The register is cleared when the value rounded to an integer is a non-zero value.

- Standard event enable register
- Extended event enable register
- Transition filter

Syntax \*PSC {<Nrf>}  
\*PSC?  
<Nrf>=0(not clear), non-zero (clear)  
Example \*PSC 1  
\*PSC? -> 1  
Description For details on the registers, see chapter 7.

### **\*RST (ReSeT)**

Function Collectively initializes the current settings of the following command groups.  
ACCumulate, ACQuire, CHANnel<x>, TIMEbase, TRIGger  
Syntax \*RST  
Example \*RST  
Description Also clears \*OPC and \*OPC? commands that have been sent earlier.

### **\*SRE (Service Request Enable register)**

Function Sets the service request enable register or queries the current setting.  
Syntax \*SRE <Nrf>  
\*SRE?  
<Nrf>=0 to 255  
Example \*SRE 239  
\*SRE? -> 175  
Description

- Specify the value as a sum of decimal values of each bit.
- For example, specifying “\*SRE 239” will cause the service request enable register to be set to “11101111.” In this case, bit 4 of the service request enable register is disabled which means that bit 4 (MAV) of the status byte register is not set to “1,” even if “the output queue is not empty.”
- Bit 6 (MSS) of the status byte register is the MSS bit itself, and therefore, is ignored.
- The default value is “\*SRE 0” (all bits disabled).
- A query using \*SRE? will not clear the contents of the service request enable register.
- For details on the service request enable register, see page 7-1.

### **\*STB? (STatus Byte)**

Function Queries the status byte register.  
Syntax \*STB?  
Example \*STB? -> 4  
Description

- The sum of the bits is returned as a decimal value.
- Since the register is read without executing serial polling, bit 6 is a MSS bit not RQS.
- For example, if a value of “4” is returned, this indicates that the status byte register is set to “00000100.” In this case, you can see that “the error queue is not empty” (an error occurred).
- A query using \*STB? will not clear the contents of the status byte register.
- For details on the status byte register, see page 7-2.

### **\*WAI (WAIt)**

Function Holds the subsequent command until the completion of the specified overlap operation.  
Syntax \*WAI  
Example \*WAI  
Description

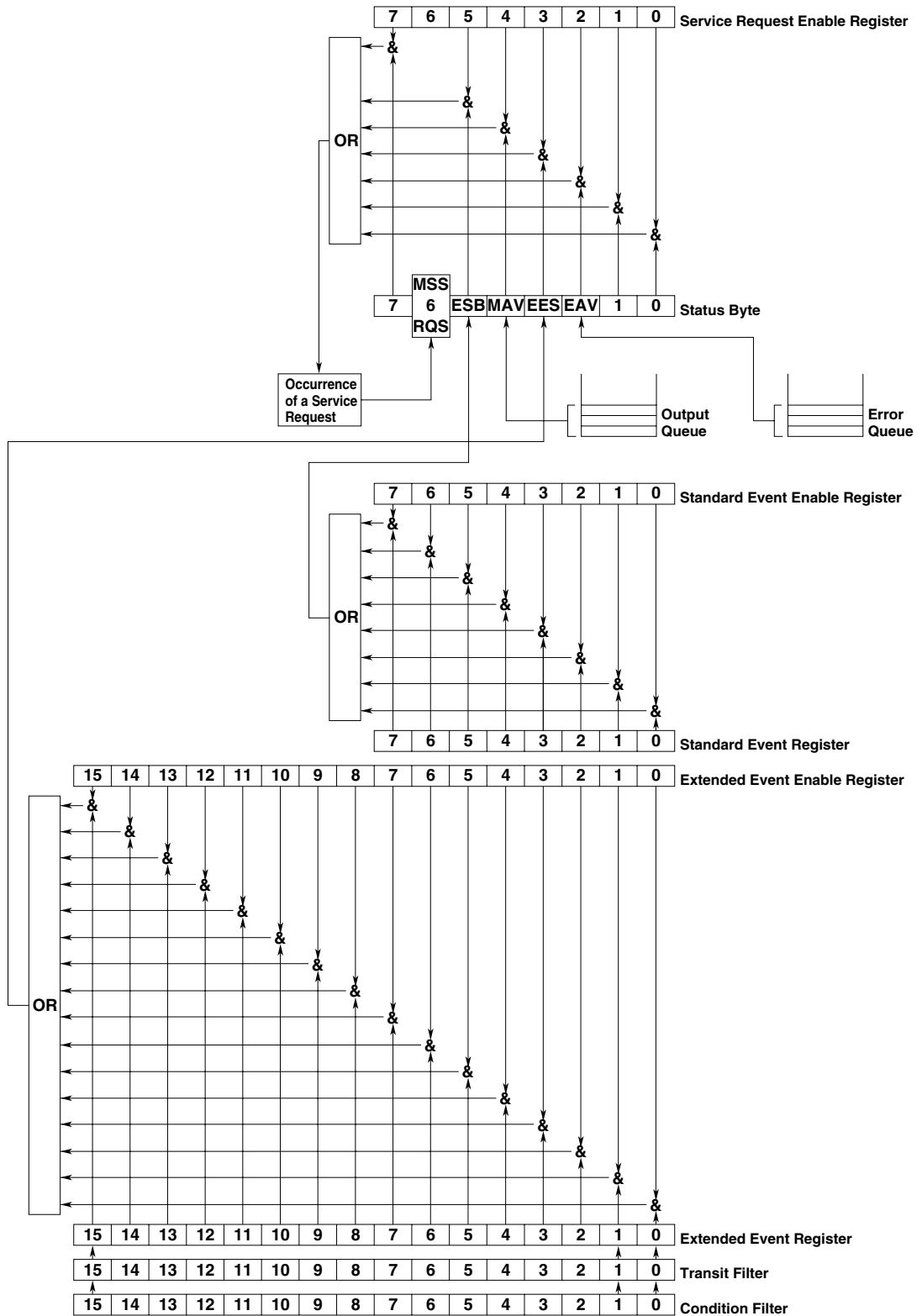
- For the description regarding how to synchronize the program using \*WAI, see page 5-7.
- The :COMMunicate:OPSE command is used to specify the overlap command.

# Chapter 7 Status Report

## 7.1 Overview of the Status Report

### 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 488.2-1992.





### 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 query 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 Query 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.

## 7.3 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 6.

#### 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 6.

## 7.3 Standard Event Register/7.4 Extended Event Register

### Operation of the Standard Event Register

The standard event register is provided for eight different kinds of event which can occur inside the instrument. Bit 5 (ESB) of the status byte is set to "1" when any of the bits in this register becomes "1" (or when the corresponding bit of the standard event enable register becomes "1").

Examples

1. A query error occurs.
2. Bit 2 (QYE) is set to "1".
3. Bit 5 (ESB) of the status byte is set to "1" if bit 2 of the standard event enable register is "1".

It is also possible to check what type of event has occurred inside the instrument by reading the contents of the standard event register.

### Reading from the Standard Event Register

The contents of the standard event register can be read by the \*ESR command. After completion of the read-out, the register will be cleared.

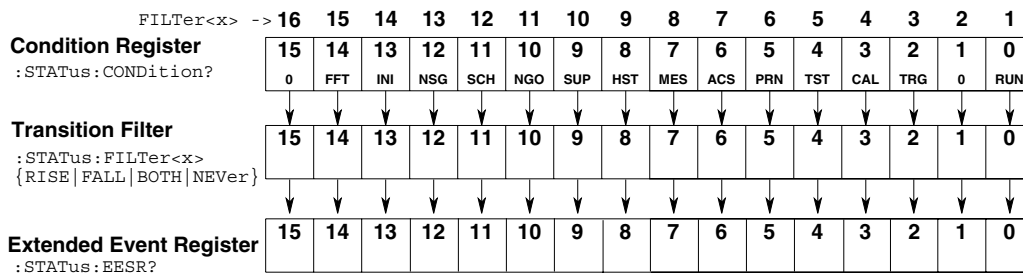
### Clearing the Standard Event Register

The standard event register is cleared in the following three cases.

- When the contents of the standard event register are read using \*ESR?
- When the \*CLS command is received
- When power is turned ON again

## 7.4 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 waiting for 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)	Sets to "1" while floppy disk drive/Zip drive, or external SCSI device is being accessed.
Bit 7 MES (Measuring)	Set to "1" when automated measurement of waveform parameters or cursor measurement is in progress.
Bit 8 HST (History Search)	Set to "1" while executing history search.
Bit 9 SUP (Set-up)	Set to "1" during auto set-up.
Bit 10 NGO (Go/No-Go)	Set to "1" while performing GO/NO-GO determination.
Bit 11 SCH (Search)	Set to "1" while executing edge/pattern search.
Bit 12 NSG (N-Single)	Set to "1" during continuous acquisition when the trigger mode is set to single (N).
Bit 13 INI (Initializing)	Sets to "1" during initialization.
Bit 14 FFT	Set to "1" while performing the FFT.

The filter is applied to each bit of the condition register separately, and can be selected from the following.

Note that the numbering of the bits used in the filter setting differs from the actual bit number (1 to 16 vs. 0 to 15).

Rise	The bit of the extended event register becomes "1" when the bit of the condition register changes from "0" to "1".
Fall	The bit of the extended event register becomes "1" when the bit of the condition register changes from "1" to "0".
Both	The bit of the extended event register becomes "1" when the bit of the condition register changes from "0" to "1", or from "1" to "0".
Never	The bit of the extended event register is disabled and always "0".

## 7.5 Output Queue and Error Queue

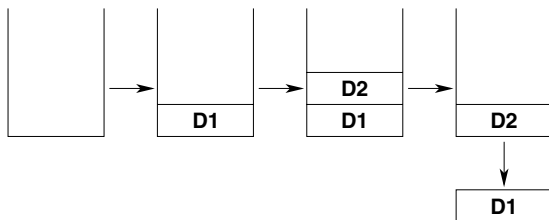
### Overview of the Output Queue

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 5-2)
- 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, if the controller sends an incorrect program message, the number, "113, "Undefined header"," and the error message are stored in the error queue, when the error is displayed. 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.

# Chapter 8 Sample Program

## 8.1 Before Programming

### Environment

Model: MS-DOS/V Computer  
Language: Visual BASIC Ver5.0 Professional Edition or more  
GP-IB board: AT-GPIB/TNT IEEE-488.2 board from National Instruments.

### Setting up the Visual Basic

Component: MSComm  
Standardmodule: Niglobal.bas  
Vbib-32.bas

### Setting up the DL750

#### GP-IB

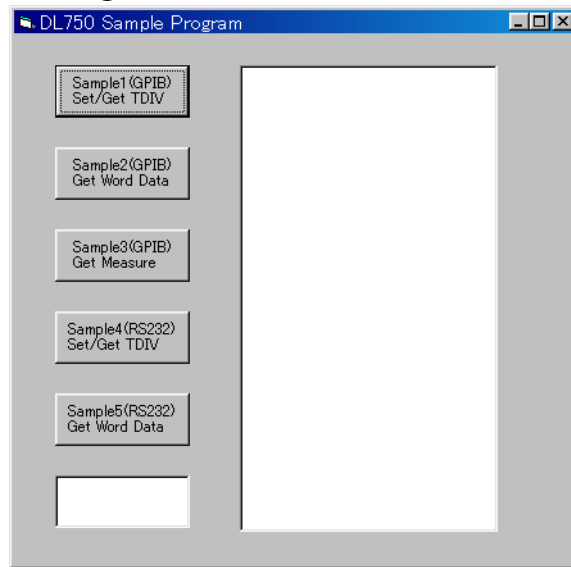
All the sample programs given in this chapter use address 1 for the DL750, so be sure to assign the instrument to address 1 as described on page 1-6.

#### Serial (RS-232)

All the sample programs in this chapter assume the below settings, so be sure to set parameters as described on page 2-8.

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

## 8.2 Image of Sample Program





## 8.3 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 = &HCOA                    '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 = "19200,N,8,1"             'Rate = 19200, 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"
        End If
    End If
End Sub

```

### 8.3 Initialize/Error/Execute

```
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)
    ElseIf (sts = 2) Then
        List1.AddItem "Waiting for Trigger!"
    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
-----

```

## 8.4 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
MSCComm1.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 MSCComm1.OutBufferCount = 0

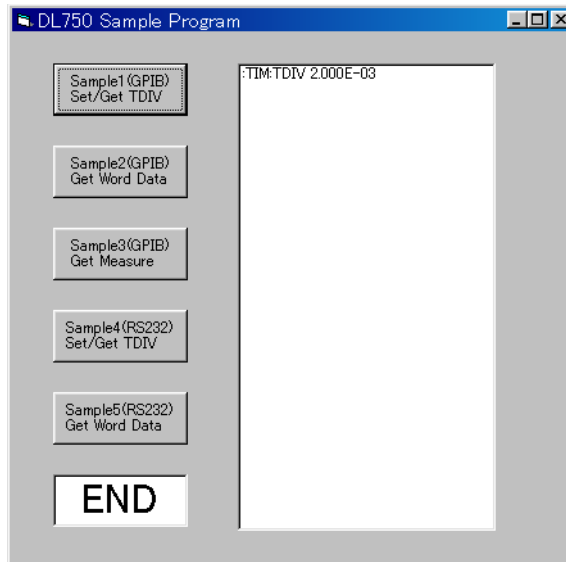
TimerCount = 1
Do                                       'Receive Query
  qry = qry + MSCComm1.Input           'Wait until End Data = Terminator
  Dummy = DoEvents()
  If (TimerCount >= Timeout) Then
    Call DisplayRS232Error("Receive Timeout", msg)
    RS232Tdiv = 1
    GoTo finish
  End If
Loop Until Right$(qry, 1) = term

Query(0) = Left$(qry, Len(qry) - 1)
RS232Tdiv = 0
finish:
MSCComm1.PortOpen = False             'Port Close
Timer1.Interval = 0

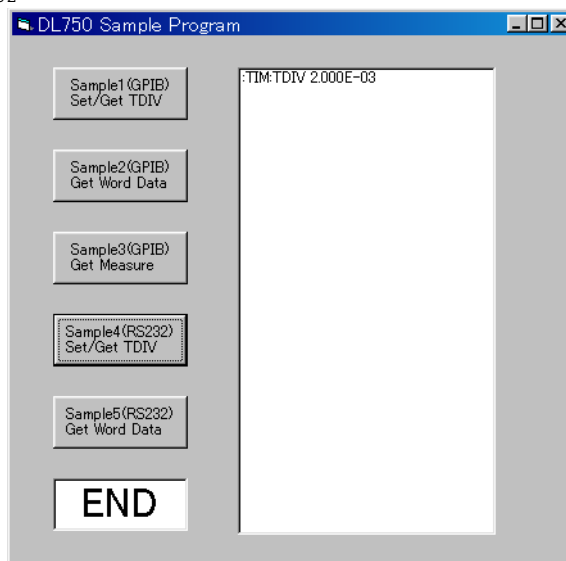
```

End Function

GP-IB



RS-232



## 8.5 Data Output in Word Format (2 bytes)

---

### 8.5 Data Output in Word Format (2 bytes)

```
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 ofs As Variant          'Offset value
    Dim eos As Integer          'EOS
    Dim hlg As Integer          'Block Header Length
    Dim dlq As Integer          'Block 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:RECORD 0" + term 'Record number = 0
    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 1001" + term 'START 0,END 1001 (Length = 1002)
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetWord = 1
        Exit Function
    End If
    msg = "WAVEFORM:RANGE?" + 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)
    msg = "WAVEFORM:OFFSET?" + term 'Get Offset 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
ofs = 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 = ilrldi(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 / 2400 + ofs
  List1.AddItem CStr(i) + ":" + CStr(dat)
Next i

eos = &HCOA
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 ofs As Variant 'Offset 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

```

## 8.5 Data Output in Word Format (2 bytes)

---

```
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")
            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:RECORD 0" + term     'Record number = 0
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
```



```

    End If
Loop Until MSComm1.OutBufferCount = 0

msg = "WAVEFORM:START 0;END 1001" + term           'START 0,END 1001 (Length = 1002)
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

qry = CStr(Empty)
msg = "WAVEFORM:RANGE?" + 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)

qry = CStr(Empty)
msg = "WAVEFORM:OFFSET?" + term                  'Get Offset 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
ofs = 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

```

## 8.5 Data Output in Word Format (2 bytes)

---

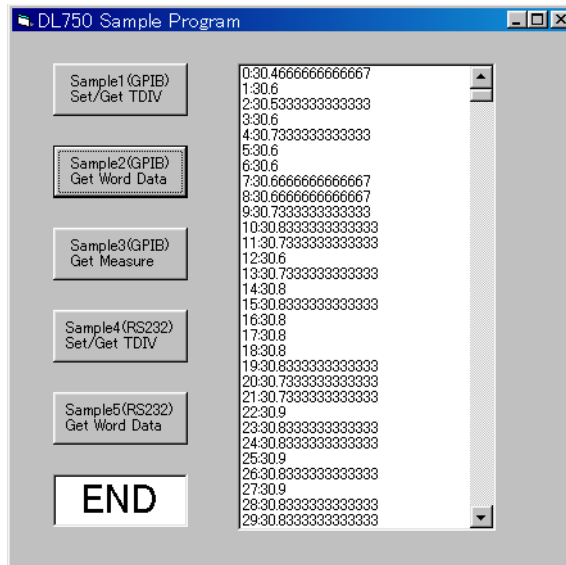
```
Dummy = DoEvents()
If (TimerCount >= Timeout) Then
    Call DisplayRS232Error("Receive Timeout", msg)
    RS232GetWord = 1
    GoTo finish
End If
Loop
qry = MSComm1.Input
dlg = Val(Left$(qry, hlg)) 'leng% = Data Byte Length
MSComm1.InputMode = comInputModeBinary 'Receive Waveform Data(2 Byte)
MSComm1.InputLen = 2
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 / 2400 + ofs
    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

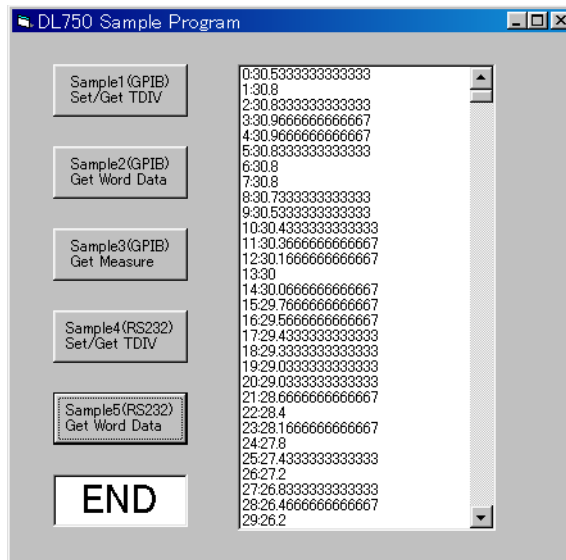
RS232GetWord = 0
finish:
MSComm1.PortOpen = False 'Port Close
Timer1.Interval = 0

End Function
-----
```

GP-1B



RS-232



## 8.6 Sets/Queries Measure Value

```

Sample3(GPIB) Get Measure value
-----
Private Function GpibGetMeasure() As Integer
    Dim msg As String           'Command buffer
    Dim qry As String           'Query biffer
    Dim sts As Integer
    Dim stp As Integer

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

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

    msg = "STOP" + term          'Acquisition = Stop
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        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 = "CHANNEL:DISPLAY ON" + term 'CH1 On
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    msg = "CHANNEL:PROBE 10" + term 'CH1 Probe = 10:1
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    msg = "CHANNEL:VDIV 500mV" + term 'CH1 V/div = 500mV
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    msg = "ACQUIRE:MODE NORMAL;RLENGTH 1000" + term 'Acquisition mode = NORMAL, length = 1000
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    msg = "TIMEBASE:TDIV 100ms" + term 'T/div = 100ms
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    msg = "TRIGGER:SIMPLE:LEVEL 500mV" + term 'Trigger level = 500mV
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If
    msg = "MEASURE:CHANNEL1:PTOPEAK:STATE ON" + term 'Measure P-P On
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GpibGetMeasure = 1
        Exit Function
    End If

```

```

msg = "MEASURE:CHANNEL1:AVERAGE:STATE ON" + term           'Measure Average On
sts = ilwrt(Dev, msg, Len(msg))
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GpibGetMeasure = 1
  Exit Function
End If
msg = "MEASURE:CHANNEL1:FREQUENCY:STATE ON" + term         'Measure Frequency On
sts = ilwrt(Dev, msg, Len(msg))
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GpibGetMeasure = 1
  Exit Function
End If
msg = "MEASURE:TRANGE -5,5" + term                         'Measure Time Range -5,5
sts = ilwrt(Dev, msg, Len(msg))
If (sts < 0) Then
  Call DisplayGPIBError(sts, msg)
  GpibGetMeasure = 1
  Exit Function
End If

msg = "SSTART? 500" + term                                 'Start Single Trigger
sts = ilwrt(Dev, msg, Len(msg))                            'Wait until stop Acquisition
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
stp = Val(qry)
If (stp = 1) Then
  GpibGetMeasure = 2
  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:CHANNEL1:PTOPEAK:VALUE?" + 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:CHANNEL1:AVERAGE:VALUE?" + term           'Get Average 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

```

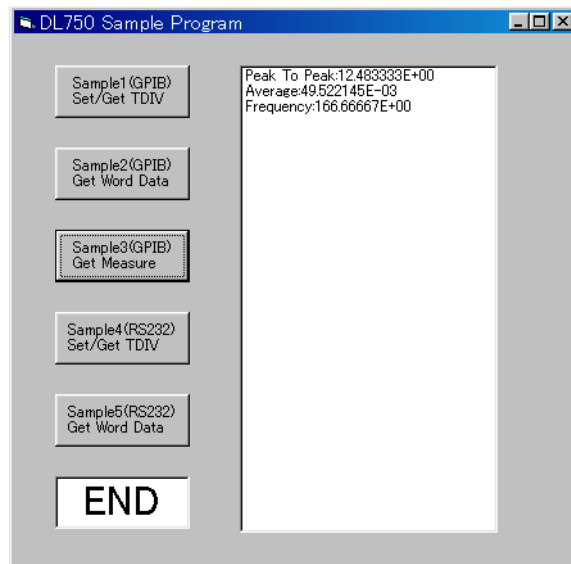
## 8.6 Sets/Queries Measure Value

```
Query(1) = "Average:" + Left$(qry, ibcntl - 1)

msg = "MEASURE:CHANNEL1:FREQUENCY:VALUE?" + 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(2) = "Frequency:" + 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
```

---



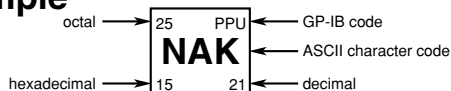
# Appendix

## Appendix 1 ASCII Character Code

ASCII character codes are given below.

	0	1	2	3	4	5	6	7
0	0 NUL	20 DEL	40 SP	0 60 0	16 @	100 P	0 120 '	160 p
1	1 SOH	21 DC1	41 !	1 61 1	17 A	101 Q	1 121 a	171 q
2	2 STX	22 DC2	42 "	2 62 2	18 B	102 R	2 122 b	182 r
3	3 ETX	23 DC3	43 #	3 63 3	19 C	103 S	3 123 c	193 s
4	4 EOT	24 DC4	44 \$	4 64 4	20 D	104 T	4 124 d	204 t
5	5 ENQ	25 NAK	45 %	5 65 5	21 E	105 U	5 125 e	215 u
6	6 ACK	26 SYN	46 &	6 66 6	22 F	106 V	6 126 f	226 v
7	7 BEL	27 ETB	47 ,	7 67 7	23 G	107 W	7 127 g	237 w
8	10 BS	30 CAN	50 (	8 70 8	24 H	110 X	8 130 h	248 x
9	11 HT	31 EM	51 )	9 71 9	25 I	111 Y	9 131 i	259 y
A	12 LF	32 SUB	52 *	10 72 :	26 J	112 Z	10 132 j	260 z
B	13 VT	33 ESC	53 +	11 73 ;	27 K	113 [	11 133 k	271 {
C	14 FF	34 FS	54 ,	12 74 <	28 L	114 \	12 134 l	282 
D	15 CR	35 GS	55 -	13 75 =	29 M	115 ]	13 135 m	293 }
E	16 SO	36 RS	56 .	14 76 >	30 N	116 ^	14 136 n	304 ~
F	17 SI	37 US	57 /	15 77 ?	UNL O	117 _	15 137 o	177 DEL (RUBOUT)
	Address Command	Universal Command	Listener Address	Talker Address	Secondary Command			

### Example



## Appendix 2 Error Messages

Error messages related to communications are given below.

- The instrument allows error messages to be displayed in either Japanese or English, however, they are shown only in English when they are displayed on a personal computer.
- When servicing is required, contact your nearest YOKOGAWA representative, given on the back cover of this manual.
- Only error messages relating to communications are given. For other error messages, refer to the User's Manual IM 701210-01E.

### Errors in Communication Command (100 to 199)

Code	Message	Action	Reference Page
102	Syntax error.	Incorrect syntax.	Chapter 5, 6
103	Invalid separator.	Insert a comma between data items to separate them.	51
104	Data type error.	Refer to pages 5-5 to 5-6 and enter using the correct data format.	55 to 56
105	GET not allowed.	GET is not supported as a response to an interface message.	17
108	Parameter not allowed.	Check the number of parameters.	5-5, Chapter 6
109	Missing parameter.	Enter required parameters.	5-5, Chapter 6
111	Header separator error.	Insert a space between header and data to separate them.	51
112	Program mnemonic too long.	Check the mnemonic (a character string consisting of letters and numbers).	Chapter 6
113	Undefined header.	Check the header.	Chapter 6
114	Header suffix out of range.	Check the header.	Chapter 6
120	Numeric data error.	Numeric value must be entered for <NR1>-format.	55
123	Exponent too large.	Use a smaller exponent for <NR3>-format.	5-5, Chapter 6
124	Too many digits.	Limit the number of digits to 255 or less.	5-5, Chapter 6
128	Numeric data not allowed.	Enter a format other than <NR1>-format.	5-5, Chapter 6
131	Invalid unit.	Check the unit for <Voltage>, <Time> and <Frequency>.	55
134	Invalid suffix.	Check the unit for <Voltage>, <Time> and <Frequency>.	55
138	Invalid suffix.	Units are allowed other than <Voltage>, <Time> and <Frequency>.	55
141	Invalid character data.	Enter one of the characters {,,,}.	Chapter 6
144	Character data too long.	Check the characters {,,,}.	Chapter 6
148	Character data not allowed.	Enter a format other than {,,,}.	Chapter 6
150	String error.	<Character string> must be enclosed by double quotation marks or single quotation marks.	56
151	Invalid string data.	<Character string> too long contains characters which cannot be used.	Chapter 6
158	String data not allowed.	Enter a data format other than <Character string>.	Chapter 6
161	Invalid block data.	<Block data> is not allowed.	5-7, Chapter 6
168	Block data not allowed.	<Block data> is not allowed.	5-7, Chapter 6
171	Invalid expression.	Equation is not allowed.	Chapter 6
178	Expression data not allowed.	Equation is not allowed.	Chapter 6
181	Invalid macro definition.	Does not conform to the macro functions specified in IEEE488.2.	—



**Error in Communication Execution (200 to 299)**

Code	Message	Action	Reference Page
221	Setpoint.	Check therelevant setting.	Chapter 6
222	Data out of range.	Check the setting range.	Chapter 6
223	Too much data.	Check the data byte length.	Chapter 6
224	Illegal parameter value.	Check the setting range.	Chapter 6
241	Hardware missing.	Check availability options.	—
260	Expression error.	Equation is not allowed.	—
270	Macro error.	Does not conform to the macro functions specified in IEC 61131-3.	—
272	Macro execution error.	Does not conform to the macro functions specified in IEC 61131-3.	—
273	Illegal macro label.	Does not conform to the macro functions specified in IEC 61131-3.	—
275	Macro definition not found.	Does not conform to the macro functions specified in IEC 61131-3.	—
276	Macro recursion error.	Does not conform to the macro functions specified in IEC 61131-3.	—
277	Macro definition not allowed.	Does not conform to the macro functions specified in IEC 61131-3.	—
278	Macro header not found.	Does not conform to the macro functions specified in IEC 61131-3.	—

**Error in Communications Query (400 to 499)**

Code	Message	Action	Reference Page
410	Query INTERRUPTED.	Check transmission/reception order.	52
420	Query UNTERMINATED.	Check transmission/reception order.	52
430	Query DEADLOCKED.	Limit the length of the program message including <PMT> to 1024 bytes or less.	52
440	Query UNTERMINATED after indefinite response.	Do not enter any query after indefinite response. *IDN and *OPT?.	—

**Error in System Operation (912)**

Code	Message	Action	Reference Page
912	Fatal error in the communication data.	Saving is required.	—

**Warning (50)**

Code	Message	Action	Reference Page
50	*OPC/ ? exists in message.	Place the *OPC or *OPC? at the end of the program message.	—

**Other Errors (350 and 390)**

Code	Message	Action	Reference Page
350	Queue overflow.	Read the error queue. Code 350 occurs when the error queue is full. This message is output only for the :STATUS:ERROR query and is not displayed on the screen.	75
390	Overrun error (only RS-232)	Execute with a lower baud rate.	—

**Note**

Code 350 indicates overflow of error queue. This code is returned as a response to the " :STATUS:ERROR?" query; it does not appear on the screen.

## Appendix 3 Overview of IEEE 488.2-1992

The GP-IB interface provided with DL750 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 Section 1.4 "GP-IB Interface Specifications".

**2 Operation of Device when the Device is Assigned to an Address Other than Addresses 0 to 30.**

The DL750 does not allow assignment to an address other than 0 to 30.

**3 Reaction when the User Changes the Address**

The current address is changed when a new address is set using MISC. 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**

4096 bytes

**b Queries which Return Multiple Response Messages**

Refer to Chapter 6, "Command List".

**c Queries which Generate Response Data during Analysis of the Syntax**

Every query generates a response data when analysis of the syntax is completed.

**d Queries which Generate Response Data during Reception**

No query generates response data when the query is received by the controller.

**e Commands Consisting of Parameters which Restrict One Other**

Some commands, like the :CHANnel<x>:VOLTage:PROBe and VDIV, have parameters which restrict unilaterally, but no commands have parameters which restrict bilaterally.

**6 Options Included in Command Function Elements and Composite Header Elements**

Refer to Chapters 5 and 6.

**7 Buffer Size which Affects Transmission of Block Data**

During transmission of block data, the output queue is extended according to the size of the data blocks.

**8 List of Program Data Elements which can be Used in Equations, and Nesting Limit**

No equations can be used.

**9 Syntax of Response to Queries**

Refer to the description of the commands given in Chapter 6.

**10 Communications between Devices which do not Follow the Response Syntax**

Not supported.

**11 Size of Data Block of Response Data**

1 to 2004000 bytes

**12 List of Supported Common Commands**

Refer to Section 6.34 "Common Command Group".

**13 Condition of Device when Calibration is Successfully Completed**

Same as the one under which measurements are performed

**14 Maximum Length of Block Data which can be Used for Definition of \*DDT Trigger Macro**

Not supported

**15 Maximum Length of Macro Label Used in Definition of Macro, 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 Section 6.34 "Common Command Group".

**17 Size of Storage Area for Protected User Data for PUD and \*PUD?**

\*PUD and \*PUD? are not supported.

**18 Length of \*RDT and \*RDT? Resource Name**

\*RDT and \*RDT? are not supported.

**19 Change in Status Due to \*RST, \*LRN?, \*RCL and \*SAV****\*RST, \*LRN?**

Refer to Section 6.34 "Common Command Group".

**\*RCL, \*SAV**

These commands are not supported.

**20 Execution Range of Self-Test Using the \*TST?**

All the memory tests (for each internal memory) given in the Self Test menu displayed using MISC can be executed.

**21 Structure of Extended Return Status**

Refer to Chapter 7.

**22 To Find out whether Each Command is Performed in Parallel or Sequentially**

Refer to Section 5.5 "Synchronization with the Controller" and to Chapter 6.

**23 Description of Execution of Each Command**

Refer to Chapter 6 of this manual and to the User's Manual IM 701210-01E.

# Index

## Symbols

*OPC command .....	5-8
*OPC? query .....	5-8
*WAI command .....	5-7
1 cycle mode .....	6-88

## A

A -> B(n) trigger .....	6-108
A Delay B trigger .....	6-109
abbreviated form .....	5-4
accumulated display .....	6-40
ACQ data .....	6-118
ACQuire group .....	6-14
action count .....	6-54
action-on-trigger .....	6-108
address .....	1-5
address command .....	1-7
angle cursor .....	6-32
ASCII character code .....	App-1
ASETup group .....	6-16
attenuation constant .....	6-73
auto scroll (search) .....	6-91
auto setup .....	6-16
automated measurement of waveform parameters .....	6-78
automated measurement, ON/OFF of .....	6-88
averaging .....	6-15
averaging (computation) .....	6-72
averaging mode (computation) .....	6-73

## B

B>Time trigger .....	6-110
baud rate .....	2-9
binary computation .....	6-73
block data .....	5-7
brightness .....	6-41

## C

CALibrate group .....	6-17
calibration .....	6-17, 6-125
CAPTure group .....	6-18
CHANnel group .....	6-20
channel, ON/OFF of .....	6-22
character string data .....	5-6
CLEar group .....	6-26
clear trace .....	6-26
click sound .....	6-97
command .....	5-3
common command group .....	6-125
common command header .....	5-3
COMMunicate group .....	6-27
compound header .....	5-3
computation .....	6-72
condition register .....	6-94, 7-4
connector (serial) .....	2-3
cursor (X-Y display) .....	6-35
CURSor group .....	6-29
cursor measurements .....	6-32
cutoff frequency .....	6-74

## D

data .....	5-5
data compression .....	6-48
data format .....	2-7, 2-9
data, loading of .....	6-47
data transfer rate .....	1-4, 4-2
date/time .....	6-97
DCL(Device Clear) .....	1-6
dead lock .....	5-2
delay .....	6-111
delay between channels .....	6-85
display format .....	6-41
DISPlay group .....	6-38
display mode .....	6-120
distal .....	6-87
dual capture .....	6-18, 6-19

## E

Edge on A trigger .....	6-111
edge search .....	6-91
enhanced trigger .....	6-113
error message .....	App-2
error queue .....	6-94, 6-125, 7-5
EVENT group .....	6-43
event waveform .....	6-43
extended event enable register .....	6-94
extended event register .....	6-94, 7-4
extra window .....	6-41

## F

FFT computation .....	6-74
FILE group .....	6-44
file, saving of .....	6-47
filter .....	6-24, 6-74
free disk space .....	6-47
front panel .....	1-1, 2-1, 3-1, 4-1

## G

GO/NO-GO determination .....	6-52
GO/NO-GO determination, type of .....	6-53
GONogo group .....	6-49
GP-IB cable .....	1-2
GP-IB interface function .....	1-3
GP-IB interface specification .....	1-4
graticule .....	6-41
grid .....	6-41
group .....	5-3
GTL(Go To Local) .....	1-6

## Index

### H

H cursor (T-Y display) .....	6-33
H cursor (X-Y display) .....	6-35
handshaking .....	2-5
handshaking method .....	2-9
HCOPy group .....	6-57
header .....	5-4
HISTory group .....	6-61
history memory .....	6-62
hold off .....	6-111

### I

ID value .....	3-4
IFC(Interface Clear) .....	1-6
IMAGe group .....	6-64
initialization .....	6-66, 6-128
INITialize group .....	6-66
input coupling .....	6-24
intensity .....	6-41
interface message .....	1-6
interpolation method .....	6-42

### L

language .....	6-98
linear scaling .....	6-24
LLO(Local Lockout) .....	1-6
local lockout .....	6-27
log start .....	6-69
logic bit .....	6-114
logic display .....	6-33
LOGic group .....	6-67
logic waveform .....	6-68
long copy .....	6-59, 6-60
LStart group .....	6-69

### M

mail transmission .....	6-53
manual scaling .....	6-75
marker cursor (T-Y display) .....	6-33
marker cursor (X-Y display) .....	6-36
MATH group .....	6-70
MEASure group .....	6-76
measurement range .....	6-88
mesial .....	6-87
message .....	5-1
model .....	6-126
monitor display .....	6-42
motor, ON/OFF of .....	6-98
multi-line message .....	1-7

### N

network interface .....	4-1
network interface specification .....	4-2
network printer output .....	6-59

### O

offset value .....	6-118
options .....	6-128
OR trigger .....	6-111
output queue .....	7-5
overlap command .....	5-7

### P

P-P data .....	6-118
parameter determination .....	6-53
password .....	4-6
printer address .....	6-59
printer output .....	6-58
probe type .....	6-25
program message .....	5-1
proximal .....	6-87
Pulse/Rotate .....	6-15

### Q

query .....	5-4
-------------	-----

### R

realtime recording .....	6-15
record length .....	6-15
remote and local mode .....	1-3, 2-2, 3-2, 4-3
remote/local .....	6-28
REN(Remote Enable) .....	1-6
response .....	5-4, 6-27
response message .....	5-1
RS-232 interface .....	2-1

### S

sample rate .....	6-100, 6-118
save .....	6-48
scaling .....	6-75
screen image data output .....	6-65
SCSI-ID .....	6-98
SDC(Selected Device Clear) .....	1-6
search function .....	6-91
SEARch group .....	6-89
search start point .....	6-92
search type .....	6-92
sequential command .....	5-7
serial interface .....	2-1
serial interface specification .....	2-2
serial polling .....	6-95
service request enable register .....	6-128
shift points, number of .....	6-75
side panel .....	1-1, 2-1, 3-1, 4-1
sign, existence of .....	6-119
simple header .....	5-3
simple trigger .....	6-112
single start .....	6-93
SNAP group .....	6-93
snapshot .....	6-93
SPD(Serial Poll Disable) .....	1-6
SPE(Serial Poll Enable) .....	1-6
split screen .....	6-42
SStart group .....	6-93
standard event enable register .....	6-125
standard event register .....	7-3
start .....	6-93
STARt group .....	6-93
status byte .....	7-2
status byte register .....	6-128
STATus group .....	6-94
status report .....	6-94, 7-1
stop .....	6-95
STOP group .....	6-95
STP cable .....	4-4
SYSTem group .....	6-96

**T**

T<<>Time triggerT .....	6-114
T>Time trigger .....	6-114
target record .....	6-62
TCP/IP setting .....	4-6
terminator .....	2-9
thermocouple type .....	6-23
time base .....	6-15, 6-100
TIMEbase group .....	6-100
timeout time .....	4-6
timer trigger .....	6-114
TinTime trigger .....	6-114
total number of points .....	6-118
ToutTime trigger .....	6-114
transition filter .....	6-95
TRIGger group .....	6-101
trigger mode .....	6-111
trigger position .....	6-112, 6-118
trigger slope (simple trigger) .....	6-112
trigger source .....	6-113
trigger source (simple trigger) .....	6-112
trigger type .....	6-115

**U**

uni-line message .....	1-6
universal command .....	1-7
USB cable .....	3-3
USB interface .....	3-1
USB interface specification .....	3-2
user name .....	4-6
user verification .....	4-3
user-defined computation .....	6-73
UTP cable .....	4-4

**V**

V cursor (T-Y display) .....	6-34
V cursor (X-Y display) .....	6-37
V/div value .....	6-26
value measured automatically .....	6-88
vertical position .....	6-25
video output .....	6-99

**W**

waveform acquisition count .....	6-15
waveform acquisition mode .....	6-15
waveform acquisition, starting of .....	6-93
waveform acquisition, stopping of .....	6-95
waveform color .....	6-41
waveform data .....	6-117, 6-119
WAVEform group .....	6-116
waveform parameters, ON/OFF of .....	6-85
waveform zoom .....	6-123
window trigger .....	6-115

**X**

X-Y display .....	6-120
XY group .....	6-120

**Z**

zone determination .....	6-55
zone, editing of .....	6-56
zone search .....	6-62
zoom .....	6-123
zoom box .....	6-124
ZOOM group .....	6-122
zoom rate .....	6-26, 6-124
zoom target .....	6-124

**Command List**

*CAL? .....	6-125
*CLS .....	6-125
*ESE .....	6-125
*ESR? .....	6-126
*IDN? .....	6-126
*LRN? .....	6-126
*OPC .....	6-127
*OPC? .....	6-127
*OPT? .....	6-128
*PSC .....	6-128
*RST .....	6-128
*SRE .....	6-128
*STB? .....	6-128
*WAI .....	6-128
:ACquire:AVERage:COUNT .....	6-15
:ACquire:AVERage:EWEight .....	6-15
:ACquire:AVERage? .....	6-15
:ACquire:CLOCK .....	6-15
:ACquire:COUNt .....	6-15
:ACquire:MODE .....	6-15
:ACquire:PROTate .....	6-15
:ACquire:RELEngth .....	6-15
:ACquire:RTOut:ANAMing .....	6-15
:ACquire:RTOut:COMMeNt .....	6-16
:ACquire:RTOut:FILEname .....	6-16
:ACquire:RTOut:MODE .....	6-15
:ACquire:RTOut? .....	6-15
:ACquire? .....	6-15
:ASETup:ADJust .....	6-16
:ASETup:EXECute .....	6-16
:ASETup:TARGeT .....	6-16
:ASETup:UNDO .....	6-16
:ASETup? .....	6-16
:CALibrate:MODE .....	6-17
:CALibrate? .....	6-17
:CALibrate[:EXECute] .....	6-17
:CAPture:BITDisp .....	6-19
:CAPture:CAPNum .....	6-19
:CAPture:MAG .....	6-19
:CAPture:MODE .....	6-19
:CAPture:POSition .....	6-19
:CAPture:TDIV .....	6-19
:CAPture:WINDow .....	6-19
:CAPture? .....	6-19
:CHANnel<x>:DISPlay .....	6-22
:CHANnel<x>:LABel .....	6-22
:CHANnel<x>:MODUle? .....	6-22
:CHANnel<x>:TEMPerature:BURNout .....	6-23
:CHANnel<x>:TEMPerature:BWIDth .....	6-23
:CHANnel<x>:TEMPerature:COUPLing .....	6-23
:CHANnel<x>:TEMPerature:RJC .....	6-23
:CHANnel<x>:TEMPerature:SCALE .....	6-23
:CHANnel<x>:TEMPerature:TYPE .....	6-23
:CHANnel<x>:TEMPerature:UNIT .....	6-24
:CHANnel<x>:TEMPerature? .....	6-23

## Index

:CHANnel<x>:VOLTage?	6-24	:CURSor[:TY]:LOGic?	6-33
:CHANnel<x>?	6-22	:CURSor[:TY]:MARKer:DX<x>?	6-34
:CHANnel<x>[:VOLTage]:BWIDth	6-24	:CURSor[:TY]:MARKer:DY<x>?	6-34
:CHANnel<x>[:VOLTage]:COUPling	6-24	:CURSor[:TY]:MARKer:M<x>:JUMP	6-34
:CHANnel<x>[:VOLTage]:INVert	6-24	:CURSor[:TY]:MARKer:M<x>:POSition	6-34
:CHANnel<x>[:VOLTage]:LSCale:AVALue	6-24	:CURSor[:TY]:MARKer:M<x>:TRACe	6-34
:CHANnel<x>[:VOLTage]:LSCale:BVALue	6-25	:CURSor[:TY]:MARKer:M<x>? .....	6-34
:CHANnel<x>[:VOLTage]:LSCale:GETMeasure	6-25	:CURSor[:TY]:MARKer:X<x>?	6-34
:CHANnel<x>[:VOLTage]:LSCale:MODE	6-25	:CURSor[:TY]:MARKer:Y<x>?	6-34
:CHANnel<x>[:VOLTage]:LSCale:UNIT	6-25	:CURSor[:TY]:MARKer?	6-33
:CHANnel<x>[:VOLTage]:LSCale:{P1X P1Y P2X P2Y}	6-25	:CURSor[:TY]:TYPE	6-34
:CHANnel<x>[:VOLTage]:LSCale?	6-24	:CURSor[:TY]:VERTical:DX?	6-35
:CHANnel<x>[:VOLTage]:OFFSet	6-25	:CURSor[:TY]:VERTical:DY?	6-35
:CHANnel<x>[:VOLTage]:POSition	6-25	:CURSor[:TY]:VERTical:JUMP	6-35
:CHANnel<x>[:VOLTage]:PROBe	6-25	:CURSor[:TY]:VERTical:PERDt?	6-35
:CHANnel<x>[:VOLTage]:SCALe	6-26	:CURSor[:TY]:VERTical:POSition<x>	6-35
:CHANnel<x>[:VOLTage]:VARiable	6-26	:CURSor[:TY]:VERTical:TRACe	6-35
:CHANnel<x>[:VOLTage]:VDIV	6-26	:CURSor[:TY]:VERTical:X<x>?	6-35
:CHANnel<x>[:VOLTage]:ZOOM	6-26	:CURSor[:TY]:VERTical:Y<x>?	6-35
:CLEar	6-26	:CURSor[:TY]:VERTical?	6-34
:COMMunicate:HEADer	6-27	:CURSor[:TY]? .....	6-32
:COMMunicate:LOCKout	6-27	:CURSor[:TY]:DEGRee:RVALue	6-32
:COMMunicate:OPSE	6-28	:DISPlay:ACCumulate:MODE	6-40
:COMMunicate:OPSR?	6-28	:DISPlay:ACCumulate:PERSistence	6-41
:COMMunicate:OVERlap	6-28	:DISPlay:ACCumulate?	6-40
:COMMunicate:REMOte	6-28	:DISPlay:COLor:BASeColor	6-41
:COMMunicate:STATus?	6-28	:DISPlay:COLor:{BACK CAPTure}	6-41
:COMMunicate:VERBose	6-28	:DISPlay:COLor:{CHANnel<x> MATH<x> PodA PodB EVENT}	6-41
:COMMunicate:WAIT	6-28	.....	6-41
:COMMunicate:WAIT?	6-28	:DISPlay:COLor?	6-41
:COMMunicate?	6-27	:DISPlay:DECimation	6-41
:CURSor:XY:HORizontal:DY?	6-36	:DISPlay:EWINDow	6-41
:CURSor:XY:HORizontal:POSition<x>	6-36	:DISPlay:FORMat	6-41
:CURSor:XY:HORizontal:STRACe	6-36	:DISPlay:GRATICule	6-41
:CURSor:XY:HORizontal:TRACe	6-36	:DISPlay:INTENsity:{CURSor MENU GRID SNAP WAVeform	6-41
:CURSor:XY:HORizontal:Y<x>?	6-36	MARKer CAPTure BACK}	6-41
:CURSor:XY:HORizontal?	6-35	:DISPlay:INTENsity?	6-41
:CURSor:XY:MARKer:POSition	6-36	:DISPlay:INTerpolate	6-42
:CURSor:XY:MARKer:STRACe	6-36	:DISPlay:MAPPing	6-42
:CURSor:XY:MARKer:T?	6-36	:DISPlay:MONitor	6-42
:CURSor:XY:MARKer:TRACe	6-36	:DISPlay:SMAPPing:{CHANnel<x> MATH<x> PODA PODB	6-42
:CURSor:XY:MARKer:X?	6-36	EVENT}	6-42
:CURSor:XY:MARKer:Y?	6-36	:DISPlay:SMAPPing?	6-42
:CURSor:XY:MARKer?	6-36	:DISPlay:SVALue	6-42
:CURSor:XY:TYPE	6-37	:DISPlay:TLABEL	6-42
:CURSor:XY:VERTical:DX?	6-37	:DISPlay:TRANslucent	6-42
:CURSor:XY:VERTical:POSition<x>	6-37	:DISPlay?	6-40
:CURSor:XY:VERTical:STRACe?	6-37	:EVENT:DCAPture	6-43
:CURSor:XY:VERTical:TRACe	6-37	:EVENT:DISPlay	6-43
:CURSor:XY:VERTical:X<x>?	6-37	:EVENT:LABEL	6-43
:CURSor:XY:VERTical?	6-37	:EVENT:POSition	6-43
:CURSor:XY?	6-35	:EVENT:VOICe	6-43
:CURSor?	6-32	:EVENT:ZOOM	6-43
:CURSor[:TY]:DEGRee:DX?	6-32	:EVENT?	6-43
:CURSor[:TY]:DEGRee:DY?	6-32	:FILE:DELete:{ASCIi BINary BMP FLOat MEASure	6-47
:CURSor[:TY]:DEGRee:JUMP	6-32	POSTscript REALtime SETup SNAP JPEG PNG}	6-47
:CURSor[:TY]:DEGRee:POSition<x>	6-32	[ :EXECute] .....	6-47
:CURSor[:TY]:DEGRee:REFerence<x>	6-32	:FILE:DIRectory?	6-47
:CURSor[:TY]:DEGRee:TRACe	6-33	:FILE:LOAD:{BINary REALtime SETup SNAP}:ABORt	6-47
:CURSor[:TY]:DEGRee:X<x>?	6-33	:FILE:LOAD:{BINary REALtime SETup SNAP}[:EXECute]	6-47
:CURSor[:TY]:DEGRee:Y<x>?	6-33	:FILE:SAVE:ANAMing	6-48
:CURSor[:TY]:DEGRee?	6-32	:FILE:SAVE:ASCIi:INTerval	6-48
:CURSor[:TY]:HORizontal:DY?	6-33	:FILE:SAVE:BINary:COMPreSSion	6-48
:CURSor[:TY]:HORizontal:POSition<x>	6-33	:FILE:SAVE:COMMeNt	6-48
:CURSor[:TY]:HORizontal:TRACe	6-33	:FILE:SAVE:NAME	6-48
:CURSor[:TY]:HORizontal:Y<x>?	6-33	:FILE:SAVE:{ASCIi BINary FLOat MEASure}[:EXECute]	6-48
:CURSor[:TY]:HORizontal?	6-33	:FILE:SAVE:{ASCIi BINary FLOat MEASure}:ABORt	6-48
:CURSor[:TY]:LOGic:FORMat	6-33	:FILE:SAVE:{ASCIi BINary FLOat}:HISTory	6-48
:CURSor[:TY]:LOGic:ORDer	6-33	:FILE:SAVE:{ASCIi BINary FLOat}:RANGe	6-48

:FILE:SAVE:{ASCI BINary FLOat}:TRACe .....	6-48	:HISTory:ABORt .....	6-62
:FILE:SAVE:{ASCI BINary FLOat}? .....	6-48	:HISTory:Clear .....	6-62
:FILE:SAVE? .....	6-47	:HISTory:DISPlay .....	6-62
:FILE? .....	6-47	:HISTory:DMODE .....	6-62
:FILE[:DIRectory]:CDIRectory .....	6-47	:HISTory:EXECute .....	6-62
:FILE[:DIRectory]:DRIVE .....	6-47	:HISTory:RECOrd .....	6-62
:FILE[:DIRectory]:FILTer .....	6-47	:HISTory:RECOrd? MINimum .....	6-62
:FILE[:DIRectory]:FREE? .....	6-47	:HISTory:SMODE .....	6-62
:FILE[:DIRectory]:MDIRectory .....	6-47	:HISTory:TIME? .....	6-62
:FILE[:DIRectory]:PATH? .....	6-47	:HISTory:ZONE:EDIT<x>:CONDition .....	6-63
:GONogo:ABORt .....	6-52	:HISTory:ZONE:EDIT<x>:HORizontal .....	6-63
:GONogo:ACONdition .....	6-52	:HISTory:ZONE:EDIT<x>:SOURce .....	6-63
:GONogo:ACTion:BUZZer .....	6-52	:HISTory:ZONE:EDIT<x>:VERTical .....	6-63
:GONogo:ACTion:HCOPY .....	6-52	:HISTory:ZONE:EDIT<x>? .....	6-63
:GONogo:ACTion:IMAGe .....	6-52	:HISTory:ZONE:LOGic .....	6-63
:GONogo:ACTion:MAIL:COUNT .....	6-53	:HISTory:ZONE? .....	6-62
:GONogo:ACTion:MAIL:MODE .....	6-53	:HISTory? .....	6-62
:GONogo:ACTion:MAIL? .....	6-53	:IMAGe:ABORt .....	6-65
:GONogo:ACTion:SAVE .....	6-53	:IMAGe:COMMeNt .....	6-65
:GONogo:ACTion? .....	6-52	:IMAGe:COMPreSSion .....	6-65
:GONogo:COUNT? .....	6-53	:IMAGe:EXECute .....	6-65
:GONogo:EXECute .....	6-53	:IMAGe:FORMat .....	6-65
:GONogo:LOGic .....	6-53	:IMAGe:SAVE:ANAMing .....	6-65
:GONogo:MODE .....	6-53	:IMAGe:SAVE:CDIRectory .....	6-65
:GONogo:NGCount? .....	6-53	:IMAGe:SAVE:DRIVE .....	6-65
:GONogo:PARAmeter:ITEM<x>:CAUSE? .....	6-54	:IMAGe:SAVE:NAME .....	6-65
:GONogo:PARAmeter:ITEM<x>:MODE .....	6-54	:IMAGe:SAVE:PATH? .....	6-65
:GONogo:PARAmeter:ITEM<x>:TRACe .....	6-54	:IMAGe:SAVE? .....	6-65
:GONogo:PARAmeter:ITEM<x>:TYPE:<Parameter> .....	6-54	:IMAGe:SEND? .....	6-65
:GONogo:PARAmeter:ITEM<x>:TYPE? .....	6-54	:IMAGe:TONE .....	6-65
:GONogo:PARAmeter:ITEM<x>:VALue? .....	6-54	:IMAGe? .....	6-65
:GONogo:PARAmeter:ITEM<x>? .....	6-54	:INITialize:EXECute .....	6-66
:GONogo:PARAmeter? .....	6-53	:INITialize:UNDO .....	6-66
:GONogo:REMOte .....	6-54	:LOGic:{PODA PODB};{BIT1 BIT2 BIT3 BIT4 BIT5 BIT6 BIT7 BIT8}	
:GONogo:SEQUeNce .....	6-54	:DISPlay .....	6-68
:GONogo:STOPcount .....	6-55	:LOGic:{PODA PODB};{BIT1 BIT2 BIT3 BIT4 BIT5 BIT6	
:GONogo:TRANge .....	6-55	BIT7 BIT8}:LABel .....	6-68
:GONogo:WAIT? .....	6-55	:LOGic:{PODA PODB}:ALL:DISPlay .....	6-68
:GONogo:ZONE:EDIT<x>:EXIT .....	6-55	:LOGic:{PODA PODB}:BMAPPING .....	6-68
:GONogo:ZONE:EDIT<x>:PART .....	6-55	:LOGic:{PODA PODB}:DISPlay .....	6-68
:GONogo:ZONE:EDIT<x>:TRACe .....	6-56	:LOGic:{PODA PODB}:LABel .....	6-68
:GONogo:ZONE:EDIT<x>:WHOLe .....	6-56	:LOGic:{PODA PODB}:POSition .....	6-69
:GONogo:ZONE:PATTern<x>:CAUSE? .....	6-56	:LOGic:{PODA PODB}:ZOOM .....	6-69
:GONogo:ZONE:PATTern<x>:MODE .....	6-56	:LOGic:{PODA PODB}? .....	6-68
:GONogo:ZONE:PATTern<x>:SEND .....	6-56	:LOGic? .....	6-68
:GONogo:ZONE:PATTern<x>:TRACe .....	6-56	:LStArt .....	6-69
:GONogo:ZONE:PATTern<x>:ZONE .....	6-56	:LStArt? .....	6-69
:GONogo:ZONE:PATTern<x>? .....	6-56	:MATH<x>:AVERAge:CCOUNt .....	6-72
:GONogo:ZONE? .....	6-55	:MATH<x>:AVERAge:COUNT .....	6-73
:GONogo? .....	6-52	:MATH<x>:AVERAge:EWEight .....	6-73
:HCOPY:ABORt .....	6-58	:MATH<x>:AVERAge:MODE .....	6-73
:HCOPY:COMMeNt .....	6-58	:MATH<x>:AVERAge:TYPE .....	6-73
:HCOPY:DIRectioN .....	6-58	:MATH<x>:AVERAge? .....	6-72
:HCOPY:EXECute .....	6-58	:MATH<x>:BINary:{CHANnel<x> MATH<x>} .....	6-73
:HCOPY:EXTPrinter:PRINter? .....	6-59	:MATH<x>:BINary? .....	6-73
:HCOPY:EXTPrinter:SElect .....	6-59	:MATH<x>:CONStant<x> .....	6-73
:HCOPY:EXTPrinter:TONE .....	6-59	:MATH<x>:DEFine .....	6-73
:HCOPY:EXTPrinter:TYPE .....	6-59	:MATH<x>:DISPlay .....	6-73
:HCOPY:EXTPrinter? .....	6-58	:MATH<x>:EXECute .....	6-73
:HCOPY:NETPrint:TONE .....	6-59	:MATH<x>:FFT:POINT .....	6-74
:HCOPY:NETPrint:TYPE .....	6-59	:MATH<x>:FFT:WINDow .....	6-74
:HCOPY:NETPrint? .....	6-59	:MATH<x>:FFT? .....	6-74
:HCOPY:PRINter:FEED .....	6-59	:MATH<x>:FILTer<x>:BAND .....	6-74
:HCOPY:PRINter:MAG .....	6-59	:MATH<x>:FILTer<x>:CUTOFF<x> .....	6-74
:HCOPY:PRINter:RANGe .....	6-59	:MATH<x>:FILTer<x>:TYPE .....	6-74
:HCOPY:PRINter:TDIV .....	6-60	:MATH<x>:FILTer<x>? .....	6-74
:HCOPY:PRINter:TYPE .....	6-60	:MATH<x>:LABel .....	6-74
:HCOPY:PRINter? .....	6-59	:MATH<x>:MODE .....	6-74
:HCOPY? .....	6-58	:MATH<x>:MREFerence .....	6-74



## Index

:MATH<x>:OPERation .....	6-75	:SStart .....	6-93
:MATH<x>:SCALE:MODE .....	6-75	:SStart? .....	6-93
:MATH<x>:SCALE:VALue .....	6-75	:StARt .....	6-93
:MATH<x>:SCALE? .....	6-75	:STATus:CONDition? .....	6-94
:MATH<x>:SHIFt:{CHANnel<x> MATH<x>} .....	6-75	:STATus:EESe .....	6-94
:MATH<x>:UNIT .....	6-75	:STATus:EESR? .....	6-94
:MATH<x>? .....	6-72	:STATus:ERRor? .....	6-94
:MEASure:CYCLemode .....	6-88	:STATus:FILTer<x> .....	6-95
:MEASure:MODE .....	6-88	:STATus:QENable .....	6-95
:MEASure:TRANge .....	6-88	:STATus:QMESsage .....	6-95
:MEASure:WAIT? .....	6-88	:STATus:SPOLI? .....	6-95
:MEASure:{CHANnel<x> MATH<x>}:<Parameter>:STATe .....	6-88	:STATus? .....	6-94
.....	6-88	:StOP .....	6-95
:MEASure:{CHANnel<x> MATH<x>}:<Parameter>:VALue? .....	6-88	:SYStem:CLICk .....	6-97
.....	6-88	:SYStem:CLOCk:DATE .....	6-97
:MEASure:{CHANnel<x> MATH<x>}:<Parameter>? .....	6-87	:SYStem:CLOCk:MODE .....	6-97
:MEASure:{CHANnel<x> MATH<x>}:ALL .....	6-85	:SYStem:CLOCk:TIME .....	6-97
:MEASure:{CHANnel<x> MATH<x>}:COPY .....	6-85	:SYStem:CLOCk? .....	6-97
:MEASure:{CHANnel<x> MATH<x>}:DELay:MEASure:COUnT .....	6-85	:SYStem:HDMotor .....	6-98
.....	6-85	:SYStem:LANGUage .....	6-98
:MEASure:{CHANnel<x> MATH<x>}:DELay:MEASure:SLOPe .....	6-86	:SYStem:LCD:AUTO:MODE .....	6-98
.....	6-86	:SYStem:LCD:AUTO:TIME .....	6-98
:MEASure:{CHANnel<x> MATH<x>}:DELay:MEASure? ..	6-85	:SYStem:LCD:AUTO? .....	6-98
:MEASure:{CHANnel<x> MATH<x>}:DELay:REFerence:COUnT .....	6-86	:SYStem:LCD:BRIGhtness .....	6-98
.....	6-86	:SYStem:LCD:MODE .....	6-98
:MEASure:{CHANnel<x> MATH<x>}:DELay:REFerence:SLOPe .....	6-86	:SYStem:LCD? .....	6-98
.....	6-86	:SYStem:MLANguage .....	6-98
:MEASure:{CHANnel<x> MATH<x>}:DELay:REFerence:SOURCe .....	6-86	:SYStem:OVERview .....	6-98
.....	6-86	:SYStem:PACTion .....	6-98
:MEASure:{CHANnel<x> MATH<x>}:DELay:REFerence:TRACe .....	6-86	:SYStem:PStart .....	6-98
.....	6-86	:SYStem:SCSI:INITialize .....	6-98
:MEASure:{CHANnel<x> MATH<x>}:DELay:REFerence? ..	6-86	:SYStem:SCSI:INTernalid .....	6-98
:MEASure:{CHANnel<x> MATH<x>}:DELay:STATe .....	6-86	:SYStem:SCSI:OWNid .....	6-99
:MEASure:{CHANnel<x> MATH<x>}:DELay:VALue? .....	6-86	:SYStem:SCSI? .....	6-98
:MEASure:{CHANnel<x> MATH<x>}:DELay? .....	6-85	:SYStem:VIDeo .....	6-99
:MEASure:{CHANnel<x> MATH<x>}:DPRoximal:MODE ..	6-87	:SYStem? .....	6-97
:MEASure:{CHANnel<x> MATH<x>}:DPRoximal:PERCent .....	6-87	:TIMebase:SOURce .....	6-100
.....	6-87	:TIMebase:SRATe? .....	6-100
:MEASure:{CHANnel<x> MATH<x>}:DPRoximal:UNIT .....	6-87	:TIMebase:TDiv .....	6-100
:MEASure:{CHANnel<x> MATH<x>}:DPRoximal? .....	6-87	:TIMebase? .....	6-100
:MEASure:{CHANnel<x> MATH<x>}:METHod .....	6-87	:TRIGGer:ABN:COUnT .....	6-108
:MEASure:{CHANnel<x> MATH<x>}? .....	6-85	:TRIGGer:ABN? .....	6-108
:MEASure? .....	6-78	:TRIGGer:ACTion:BUZZer .....	6-108
:SEARch:ASCRoll:DIRectiOn .....	6-91	:TRIGGer:ACTion:HCOPY .....	6-108
:SEARch:ASCRoll:SPeEd .....	6-91	:TRIGGer:ACTion:IMAGe .....	6-108
:SEARch:ASCRoll? .....	6-91	:TRIGGer:ACTion:MAIL:COUnT .....	6-108
:SEARch:EDGE:COUnT .....	6-91	:TRIGGer:ACTion:MAIL:MODE .....	6-108
:SEARch:EDGE:HYSteresis .....	6-91	:TRIGGer:ACTion:MAIL? .....	6-108
:SEARch:EDGE:LEVEl .....	6-91	:TRIGGer:ACTion:MODE .....	6-108
:SEARch:EDGE:LOGic:CONDition .....	6-91	:TRIGGer:ACTion:SAVE .....	6-109
:SEARch:EDGE:LOGic:{A1 A2 A3 A4 A5 A6 A7 A8 B1 B2		:TRIGGer:ACTion:SEQuence .....	6-109
B3 B4 B5 B6 B7 B8} .....	6-91	:TRIGGer:ACTion? .....	6-108
:SEARch:EDGE:LOGic? .....	6-91	:TRIGGer:ADB:DELay .....	6-109
:SEARch:EDGE:SLOPe .....	6-92	:TRIGGer:ADB? .....	6-109
:SEARch:EDGE:SOURce .....	6-92	:TRIGGer:ATRigger:CHANnel<x> .....	6-109
:SEARch:EDGE? .....	6-91	:TRIGGer:ATRigger:CONDition .....	6-109
:SEARch:EXECute .....	6-92	:TRIGGer:ATRigger:LOGic:{A1~A8 B1~B8} .....	6-109
:SEARch:MAG<x> .....	6-92	:TRIGGer:ATRigger:LOGic? .....	6-109
:SEARch:MLINKage .....	6-92	:TRIGGer:ATRigger:{PODA PODB} .....	6-109
:SEARch:PLINKage .....	6-92	:TRIGGer:ATRigger? .....	6-109
:SEARch:POSition<x> .....	6-92	:TRIGGer:BGTime:TIME .....	6-110
:SEARch:SElect .....	6-92	:TRIGGer:BGTime? .....	6-110
:SEARch:SElect? MAXimum .....	6-92	:TRIGGer:BLTIME:TIME .....	6-110
:SEARch:SPOint .....	6-92	:TRIGGer:BLTime? .....	6-110
:SEARch:TDiv<x> .....	6-92	:TRIGGer:BTOut:TIME .....	6-110
:SEARch:TWINdow .....	6-92	:TRIGGer:BTOut? .....	6-110
:SEARch:TYPE .....	6-92	:TRIGGer:BRigger:CHANnel<x> .....	6-110
:SEARch? .....	6-91	:TRIGGer:BRigger:CONDition .....	6-111
:SNAP .....	6-93	:TRIGGer:BRigger:LOGic:{A1~A8 B1~B8} .....	6-110

:TRIGger:BTriGger:LOGic?	6-110	:WAVeform:SEND?	6-119
:TRIGger:BTriGger:{PODA PODB}	6-111	:WAVeform:SIGN?	6-119
:TRIGger:BTriGger?	6-110	:WAVeform:SRATe?	6-119
:TRIGger:DELay	6-111	:WAVeform:STARt	6-119
:TRIGger:EOA:CHANnel<x>	6-111	:WAVeform:TRACe	6-119
:TRIGger:EOA:CONDition	6-111	:WAVeform:TRIGger?	6-119
:TRIGger:EOA:{PODA PODB}	6-111	:WAVeform:TYPE?	6-120
:TRIGger:EOA?	6-111	:WAVeform?	6-117
:TRIGger:HOLDoff:TIME	6-111	:XY<x>:MODE	6-120
:TRIGger:HOLDoff?	6-111	:XY<x>:TRANge	6-120
:TRIGger:MODE	6-111	:XY<x>:XAXis	6-121
:TRIGger:OR:CHANnel<x>	6-112	:XY<x>:XTRace	6-121
:TRIGger:OR:{PODA PODB}	6-112	:XY<x>:YTRace	6-121
:TRIGger:OR?	6-111	:XY<x>?	6-120
:TRIGger:POSition	6-112	:ZOOM:ALLocation:ALL	6-123
:TRIGger:SCOUnt	6-112	:ZOOM:ALLocation:{CHANnel<x> MATH<x> PODA PODB}	6-124
:TRIGger:SIMPlE:SOURce	6-112		6-124
:TRIGger:SOURce:CHANnel<x>:CENTer	6-113	:ZOOM:ALLocation?	6-123
:TRIGger:SOURce:CHANnel<x>:HYSTerEsis	6-113	:ZOOM:FITMeasure	6-124
:TRIGger:SOURce:CHANnel<x>:LEVel	6-113	:ZOOM:FORMat	6-124
:TRIGger:SOURce:CHANnel<x>:WIDTh	6-114	:ZOOM:MAG<x>	6-124
:TRIGger:SOURce:CHANnel<x>?	6-113	:ZOOM:MLINKage	6-124
:TRIGger:SOURce:LOGic:{A1~A8 B1~B8}	6-114	:ZOOM:MODE	6-124
:TRIGger:SOURce?	6-113	:ZOOM:MOVE	6-124
:TRIGger:TGTime:TIME	6-114	:ZOOM:PLINKage	6-124
:TRIGger:TGTime?	6-114	:ZOOM:POSition<x>	6-124
:TRIGger:TIMer:DATE	6-114	:ZOOM:TDIV<x>	6-124
:TRIGger:TIMer:INTerval	6-114	:ZOOM?	6-123
:TRIGger:TIMer:TIME	6-114		
:TRIGger:TIMer?	6-114		
:TRIGger:TITime:TIME<x>	6-114		
:TRIGger:TITime?	6-114		
:TRIGger:TLTime:TIME	6-114		
:TRIGger:TLTime?	6-114		
:TRIGger:TOTime:TIME<x>	6-115		
:TRIGger:TOTime?	6-114		
:TRIGger:TYPE	6-115		
:TRIGger:WINDow:CHANnel<x>	6-115		
:TRIGger:WINDow:{PODA PODB}	6-115		
:TRIGger:WINDow?	6-115		
:TRIGger?	6-107		
:TRIGger[:SIMPlE]:HYSTerEsis	6-112		
:TRIGger[:SIMPlE]:LEVel	6-112		
:TRIGger[:SIMPlE]:LOGic:BIT	6-112		
:TRIGger[:SIMPlE]:LOGic:SLOPe	6-113		
:TRIGger[:SIMPlE]:LOGic?	6-112		
:TRIGger[:SIMPlE]:SLOPe	6-112		
:TRIGger[:SIMPlE]?	6-112		
:WAVeform:BITS?	6-117		
:WAVeform:BYTeorder	6-117		
:WAVeform:CAPTure:END	6-117		
:WAVeform:CAPTure:LENGth?	6-117		
:WAVeform:CAPTure:RECOrd	6-117		
:WAVeform:CAPTure:RECOrd? MAXimum	6-117		
:WAVeform:CAPTure:RECOrd? MINimum	6-117		
:WAVeform:CAPTure:SEND?	6-117		
:WAVeform:CAPTure:SRATe?	6-118		
:WAVeform:CAPTure:STARt	6-118		
:WAVeform:CAPTure:TRIGger?	6-118		
:WAVeform:CAPTure?	6-117		
:WAVeform:DATaselect	6-118		
:WAVeform:END	6-118		
:WAVeform:FORMat	6-118		
:WAVeform:LENGth?	6-118		
:WAVeform:MODule?	6-118		
:WAVeform:OFFSet?	6-118		
:WAVeform:RANGe?	6-118		
:WAVeform:RECOrd	6-118		
:WAVeform:RECOrd? MINimum	6-119		