signal explorer™

DL9140/DL9140L/ DL9240/DL9240L

Digital Oscilloscope Communication Interface

USER'S MANUAL

YOKOGAWA
Yokogawa Electric Corporation

IM 701310-17E
3rd Edition

Thank you for purchasing the DL9140/DL9140L/DL9240/DL9240L Digital Oscilloscope. This Communication Interface User's Manual describes the functions and commands of the following communication interfaces.

- USB Interface
- Ethernet Interface (Optional)

To ensure correct use, please read this manual thoroughly before beginning operation. After reading the manual, keep it in a convenient location for quick reference whenever a question arises during operation.

The following manuals are provided for the DL9140/DL9140L/DL9240/DL9240L. Please read all of them.

Manual Title	Manual No.	Description
DL9140/DL9140L/DL9240/DL9240L User's Manual	IM 701310-01E	Explains all functions and procedures of the DL9000 excluding the communication functions.
DL9140/DL9140L/DL9240/DL9240L Communication Interface User's Manual (CD-ROM)	IM 701310-17E	This manual. Describes the communication functions of the USB and Ethernet interfaces.

Notes

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

USB Interface and Ethernet Interface

- The items below are needed on the PC to use the communication functions via the USB interface.
  - DL Series Library (TMCTL)
  - USB device driver for connecting the PC and the DL9140/DL9140L/DL9240/DL9240L
- The items below are needed on the PC to use the communication functions via the Ethernet interface.
  - DL Series Library (TMCTL)

  The library and driver above can be downloaded from the following Web page.
  http://www.yokogawa.com/tm/tm-softdownload.htm

Sample Programs

Sample programs can be downloaded from the following Web page.
http://www.yokogawa.com/tm/tm-softdownload.htm

3rd Edition: December 2005 (YK)

Trademarks

- Microsoft, MS-DOS, Visual C++, Windows, and Windows NT are registered trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe and Acrobat are trademarks of Adobe Systems Incorporated.
- For purposes of this manual, the TM and ® symbols do not accompany their respective trademark names or registered trademark names.
- The TCP/IP software of this product and the document concerning the TCP/IP software have been developed/created by YOKOGAWA based on the BSD Networking Software, Release 1 that has been licensed from the University of California.
- Other company and product names are trademarks or registered trademarks of their respective holders.

Revisions

1st Edition: June 2005
2nd Edition: September 2005
3rd Edition: December 2005

How to Use This Manual

Structure of This Manual

This user's manual consists of the following sections.

Chapter 1 Overview of the USB Interface
Describes the functions and specifications of the USB interface.

Chapter 2 Overview of the Ethernet Interface (Optional)
Describes the functions and specifications of the Ethernet interface.

Chapter 3 Overview of the GP-IB Interface
Describes the functions and specifications of the GP-IB interface.

Chapter 4 Before Programming
Describes the syntax used to transmit commands.

Chapter 5 Command
Describes all the commands one by one.

Chapter 6 Status Reports
Describes the status byte, various registers, and queues.

Appendix
Describes reference material such as an ASCII character code table.

Conventions Used in This Manual

Safety Markings

The following markings are used in this manual.

*Improper handling or use can lead to injury to the user or damage to the instrument.*This symbol appears on the instrument to indicate that the user must refer to the user's manual for special instructions. The same symbol appears in the corresponding place in the user's manual to identify those instructions. In the manual, the symbol is used in conjunction with the word "WARNING" or "CAUTION."

WARNING Calls attention to actions or conditions that could cause serious or fatal injury to the user, and precautions that can be taken to prevent such occurrences.

CAUTION Calls attentions to actions or conditions that could cause light injury to the user or damage to the instrument or user's data, and precautions that can be taken to prevent such occurrences.

Note Calls attention to information that is important for proper operation of the instrument.

Notations Used on Pages Describing Operating Procedures

On pages that describe the operating procedures in Chapter 1 through 3, the following notations are used to distinguish the procedures from their explanations.

Procedure This subsection contains the operating procedure used to carry out the function described in the current chapter. All procedures are written with inexperienced users in mind; experienced users may not need to carry out all the steps.

Explanation This section describes the setup items and the limitations regarding the procedures.

Notations Used in the Procedures

Panel Keys and Soft keys
Bold characters used in the procedural explanations indicate characters that are marked on the panel keys or the characters of the soft keys or menus displayed on the screen.

SHIFT+Panel Key
SHIFT+key means you will press the SHIFT key to turn ON the SHIFT key followed by the operation key. The setup menu marked in purple above (or below) the panel key that you pressed appears on the screen.

Rotary knob/SET key
Rotary knob/SET key indicates selecting or setting parameters and entering values using the rotary knob, the SET key, and other keys. For details on the procedure, see section 4.1 or 4.2 in the *User's Manual IM701310-01E*.

Unit

k Denotes 1000. Example: 100 kS/s

K Denotes 1024. Example: 459 KB (file data size)

Symbols Used in the Syntax

The following table indicates symbols that are used in the syntax mainly in chapter 5. These symbols are referred to as BNF (Backus-Naur Form) symbols. For details on the data, see pages 4-5 and 4-6.

Symbol	Meaning	Example	Example of Input
<>	Defined value	CHANnel<x> <x> = 1 to 4	-> CHANNEL2
{}	Select from values given in {}	COUPling {AC|DC|DC50|GND}	-> COUPLING AC
|	Exclusive OR		
[]	Can be omitted	TRIGger [:SIMPle]:SLOPe	-> TRIGger:SLOPe

Contents

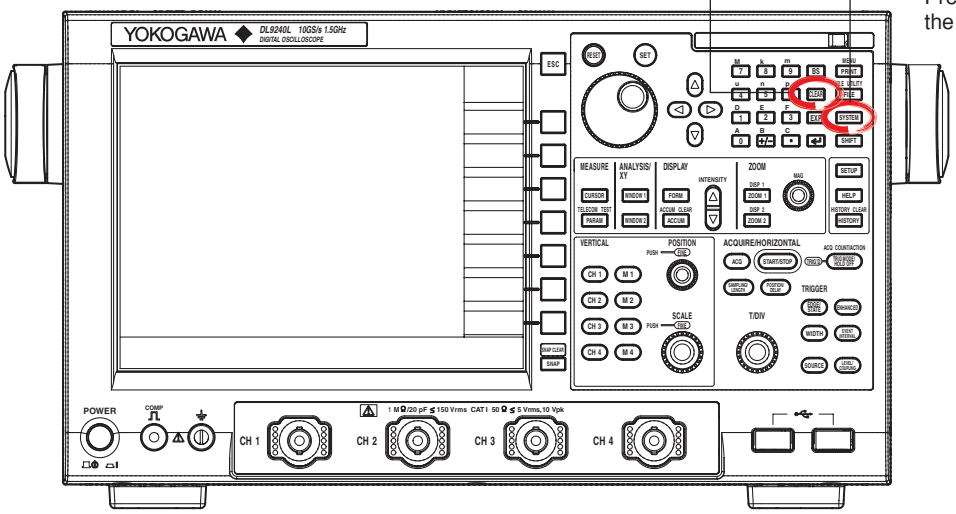
Chapter 1 Overview of the USB Interface

1.1 Part Names and Functions

Front Panel

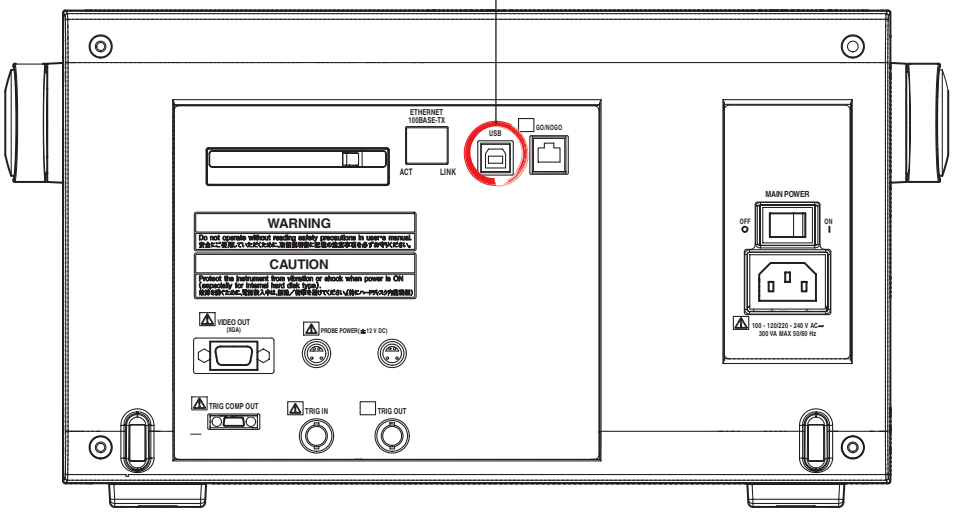
CLEAR key
Press this key to clear the remote mode (controlled via communications) and enter the local mode in which key operations are enabled. However, this act is invalid if the instrument has been set to Local Lockout mode (see page 1-2) by the controller.

SYSTEM key
Press this key to select the USB interface.



Rear Panel

USB interface connector
Connector used to connect the DL9000 to the controller (such as a PC) using a USB cable.



1.2 USB Interface Functions and Specifications

USB Interface Functions

Reception Function

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

Transmission Function

Outputs measured and computed data.
Outputs panel setup parameters and the status byte.
Outputs error codes that have occurred.

USB Interface Specifications

Electrical and mechanical specifications: Conforms to USB Rev. 2.0

Connector:	Type B connector (receptacle)
Number of ports:	1
Power supply:	Self-powered
PC system supported:	PC running Windows 2000 or Windows XP with a standard USB port (a separate device driver is needed to connect to a PC).

Data Rate

The response time when outputting waveform data is shown below as a reference.

Model:	701313 main unit
Controller:	PC (Pentium4 3.4 GHz, USB2.0) and OS (Windows XP Professional SP1)
Language used:	Visual C++

Number of Data Points	Word Data	ASCII Data
2500	Approx. 51 ms	Approx. 0.469 s
125000	Approx. 193 ms	Approx. 22.766 s
1250000	Approx. 1606 ms	Approx. 224.890 s
2500000	Approx. 3188 ms	Approx. 451.297 s
6250000	Approx. 7841 ms	Approx. 1127.625 s

Switching between Remote and Local Modes

When Switching from Local to Remote Mode

If the DL9000 receives an REN (Remote Enable) message from the PC when it is in the local mode, it switches to the remote mode.

- REMOTE is displayed in the center of the upper section of the screen.
- All keys except the **CLEAR** key are disabled.
- Settings entered in local mode are retained even when the DL9000 switches to remote mode.

When Switching from Remote to Local Mode

Pressing **CLEAR** in remote mode puts the instrument in local mode. However, this is void.

- REMOTE indication in the center of the upper section of the screen disappears.
- Key operations are enabled.
- Settings entered in remote mode are retained even when the DL9000 switches to local mode.

Note

The USB interface cannot be used simultaneously with the Ethernet interface.

1.3 Connection via the USB Interface

Precautions to Be Taken When Making Connections

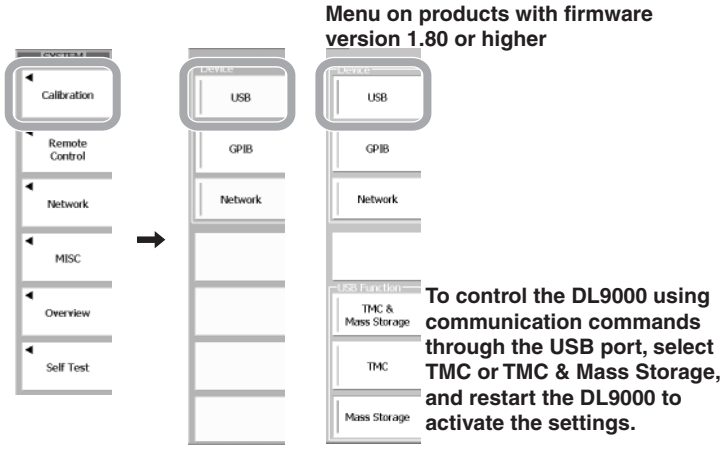
- Connect the USB cable by inserting the connector firmly into the USB connector.
- When connecting multiple devices using USB hubs, connect the DL9000 to the USB hub that is closest to the controller.
- Do not insert the USB cable into the GO/NO-GO output terminal. If you do, the instrument may malfunction.

1.4 Setting the DL9000 (USB)

Procedure

Selecting the Communication Interface

1. Press **SYSTEM**.
2. Press the **Remote Control** soft key.
3. Press the **USB** soft key.



Note

- Only the communication interface selected by Remote Control is enabled. The DL9000 does not accept commands that are transmitted to other unselected communication interfaces.
- To control the DL9000 remotely using communication commands through the USB port, select “TMC” in the menu above, and carry out the procedure below.
  - You must restart the DL9000 to activate the TMC, Mass Storage, or TMC & Mass Storage setting. Wait at least 10 seconds after you turn the power switch OFF, and then turn the switch back ON.
  - Install YOKOGAWA’s TMC (Text and Measurement Class) driver into your PC. To obtain YOKOGAWA’s USB TMC driver, contact your nearest YOKOGAWA dealer or access the following USB driver page at our Web site and download it.
    http://www.yokogawa.com/tm/tm-softdownload.htm
  - Only use the USB TMC driver (or software) provided by YOKOGAWA.
- The DL9000 can be controlled remotely even when TMC & Mass Storage is enabled and the DL9000 is connected to the PC. However, file operation is not possible using communication commands. In addition, file operation is also not possible using the keys on the DL9000. Disconnect the PC or activate the TMC setting and connect the DL9000 to the PC. File operation is also not possible using keys on the DL9000 when the Mass Storage setting is enabled.

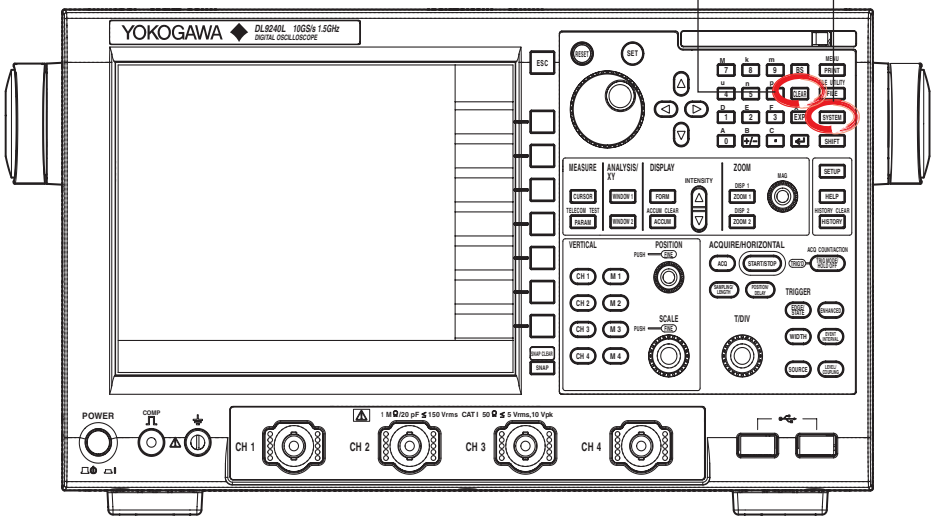
Chapter 2 Ethernet Interface

2.1 Part Names and Functions

Front Panel

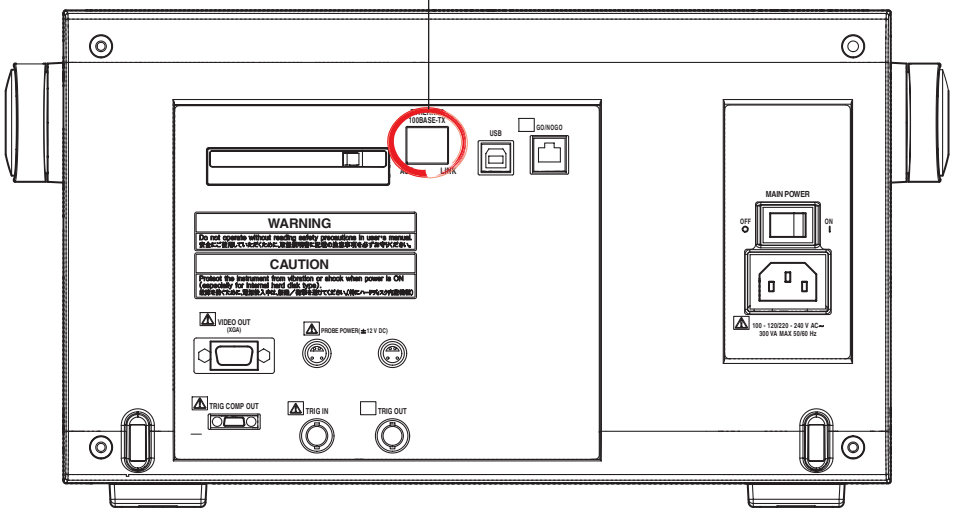
CLEAR key
Press this key to clear the remote mode (controlled via communications) and enter the local mode in which key operations are enabled. However, this act is invalid if the instrument has been set to Local Lockout mode (see page 1-2) by the controller.

SYSTEM key
Press this key to select the Ethernet interface.



Rear Panel

Ethernet interface connector
Connector used to connect the DL9000 to the controller (such as a PC) using an Ethernet cable.



2.2 Ethernet Interface Functions and Specifications

Ethernet Interface Functions

Reception Function

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

Transmission Function

Outputs measured and computed data.
Outputs panel setup parameters and the status byte.
Outputs error codes that have occurred.

Ethernet Interface Specifications

Electrical and mechanical specifications: Conforms to IEEE802.3
Number of simultaneous connections: 1
Port No.: 10001/tcp

Data Rate

The response time when outputting waveform data is shown below as a reference.
Model: 701313 main unit
Controller: PC (Pentium4 3.4 GHz) and OS (Windows XP Professional SP1)
Network adapter: Corega FEther PCI-TXL
Language used: Visual C++

Number of Data Points	Word Data	ASCII Data
2500	Approx. 53 ms	Approx. 0.391 s
125000	Approx. 259 ms	Approx. 19.063 s
1250000	Approx. 2313 ms	Approx. 189.812 s
2500000	Approx. 4595 ms	Approx. 379.750 s
6250000	Approx. 10400 ms	Approx. 950.532 s

Switching between Remote and Local Modes

When Switching from Local to Remote Mode

If the DL9000 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 in the center of the upper section of the screen.
- All keys except the **CLEAR** key are disabled.
- Settings entered in local mode are retained even when the DL9000 switches to remote mode.

When Switching from Remote to Local Mode

Pressing **CLEAR** in remote mode puts the instrument in local mode. However, this is void when the DL9000 has received a ":COMMunicate:LOCKout ON" command from the PC (local lockout condition). When the DL9000 receives a ":COMMunicate:REMote OFF" command from the PC, the DL9000 switches to the local mode regardless of the local lock condition.

- REMOTE indication in the center of the upper section of the screen disappears.
- Key operations are enabled.
- Settings entered in remote mode are retained even when the DL9000 switches to local mode.

Note

The Ethernet interface cannot be used simultaneously with the USB interface.

User Authentication Function

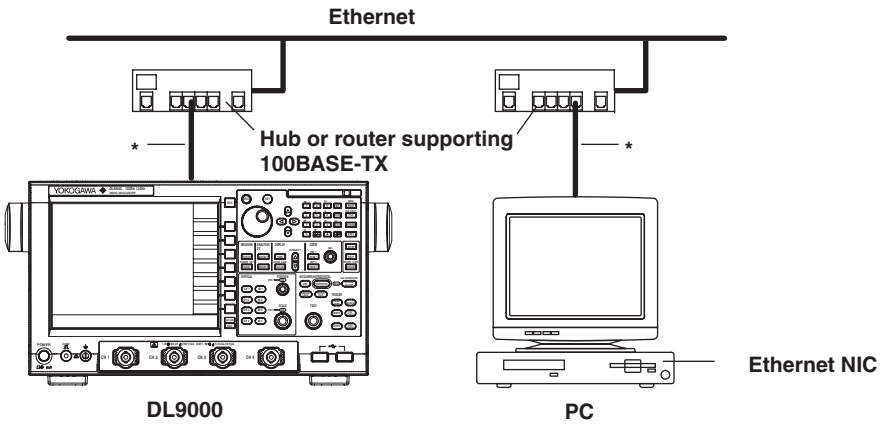
When using the Ethernet interface, a user name and password are required when connecting to the network. If the DL series library (TMCTL) version is 1.40 or later, the password is encrypted with an MD5 algorithm (RSA Data Security, Inc. MD5 Message-Digest Algorithm) and sent to the DL9000.

The user name and password are set on the Remote Control setup screen under the SYSTEM menu of the DL9000. For the setup procedure, see section 2.4, “Setting the DL9000 (Network).”

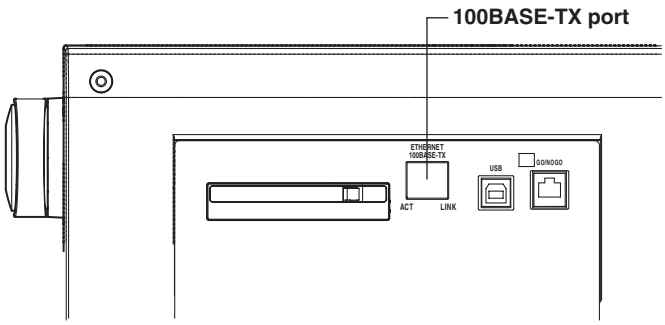
2.3 Connecting the Ethernet Interface

Connection Procedure

Connect a UTP (Unshielded Twisted-Pair) cable or an STP (Shielded Twisted-Pair) cable that is connected to a hub, for example, to the 100BASE-TX port on the rear panel of the DL9000.



* UTP cable or STP cable (straight cable in either case)



Precautions to Be Taken When Making Connections

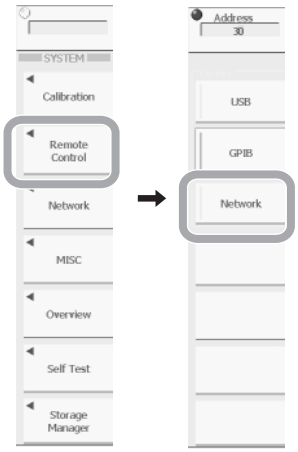
- Be sure to use a straight cable via a hub for the connection between the DL9000 and the PC. Operation is not guaranteed when the DL9000 and the controller are connected one-to-one using a cross cable.
- When using a UTP cable (straight cable), use a cable of category 5.

2.4 Setting the DL9000 (Network)

Procedure

Selecting the Communication Interface

1. Press **SYSTEM**.
2. Press the **Remote Control** soft key. The Device menu appears.
3. Press the **Network** soft key.

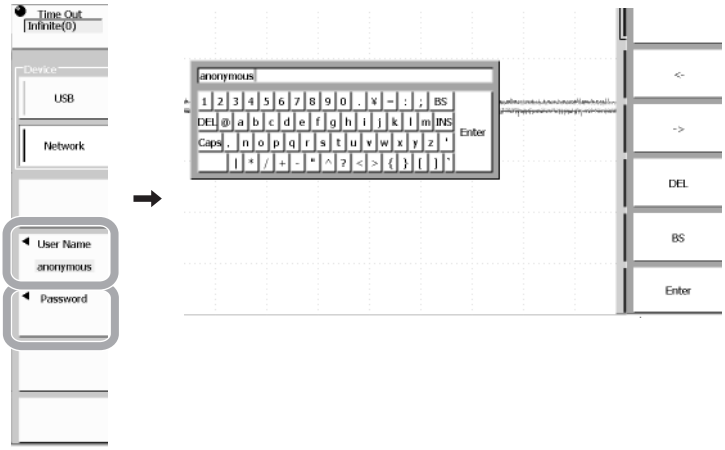


Note

Only the communication interface selected by Remote Control is enabled. The DL9000 does not accept commands that are transmitted to other unselected communication interfaces.

User Name and Password

4. Press the **UserName** soft key. A keyboard appears.
5. Enter the user name according to the procedures given in section 4.2 in the *DL9140/DL9140L/DL9240/DL9240L User's Manual (IM701310-01E)*.
6. Likewise, enter the password.



Setting the Timeout Value

7. Set the timeout value using the **rotary knob**.

Set the timeout value

Note

- For details on the keyboard (software keyboard) operation, see section 4.2, "Entering Values and Character Strings" in the *DL9140/DL9140L/DL9240/DL9240L User's Manual (IM701310-01E)*.
- User name and password are case-sensitive.

Setting the TCP/IP Parameters

To use the Ethernet interface function, the following TCP/IP settings must be entered.

- IP address
- Subnet mask
- Default gateway

For details on how to enter these parameters, see section 15.2, "Setting the TCP/IP Parameters" in the *DL9140/DL9140L/DL9240/DL9240L User's Manual (IM701310-01E)*.

Explanation

Enter the following settings when using a controller to set information that can be specified through key operation on the DL9000 or when outputting setting parameters or output waveform data to the controller.

Setting the User Name and Password

The Ethernet interface has a user verification function. A user name and password for the DL9000 are set in advance.

- **Setting the User Name**
  Enter the user name using up to 30 characters. The default setting is "anonymous."
- **Setting the Password**
  Enter the password using up to 30 characters.

Setting the Timeout Value

The connection to the DL9000 is automatically dropped if there is no access to the DL9000 for the specified time.

Setting the TCP/IP Parameters

For details, see section 15.2, "Setting the TCP/IP Parameters" in the *DL9140/DL9140L/DL9240/DL9240L User's Manual (IM701310-01E)*.

Note

- If the user authentication fails, connection to the DL9000 is dropped.
- A password is not required if the user name is "anonymous."

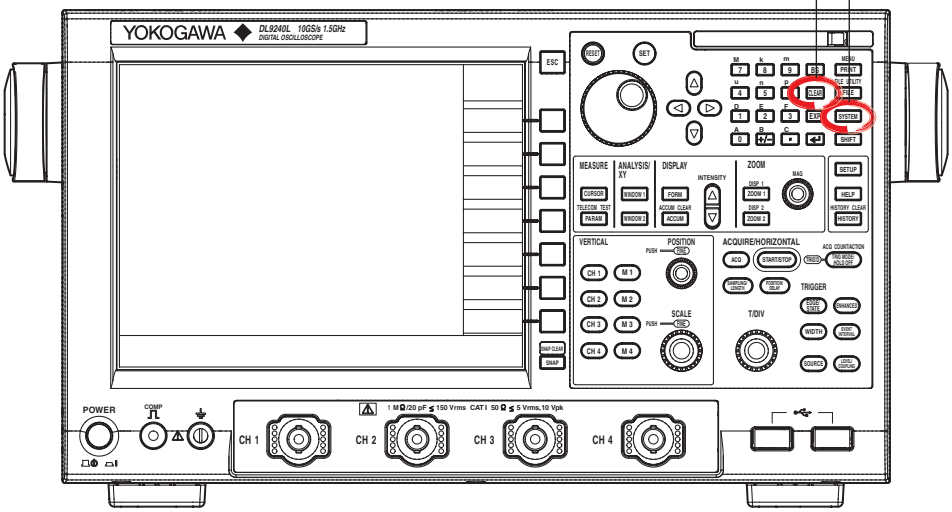
Chapter 3 Overview of the GP-IB Interface

3.1 Part Names and Functions

Front Panel

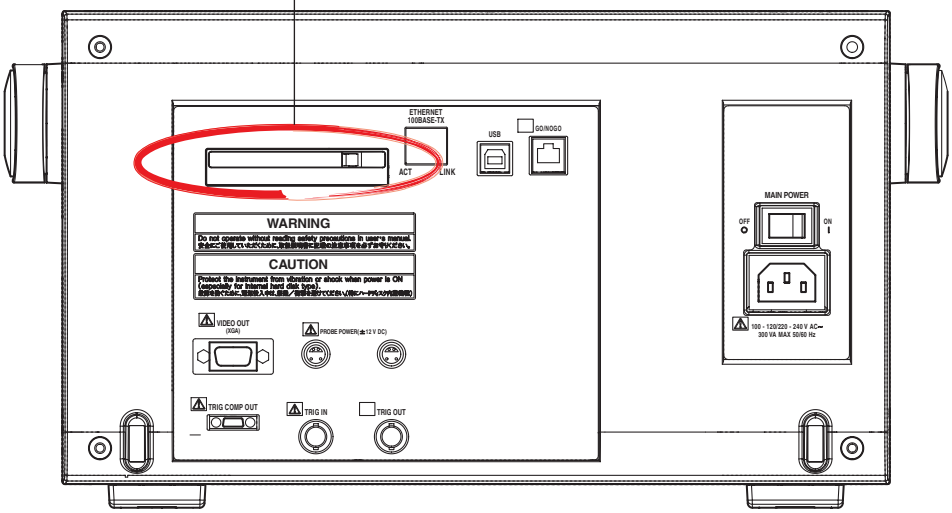
CLEAR key
Press this key to clear the remote mode (controlled via communications) and enter the local mode in which key operations are enabled. However, this act is invalid if the instrument has been set to Local Lockout mode (see page 1-2) by the controller.

SYSTEM key
Press this key to select the GP-IB interface.



Rear Panel

PC card slot
Connect a GP-IB card to the PC card slot and connect to a controller (such as a PC).



3.2 Connecting the GP-IB Card

GP-IB Card

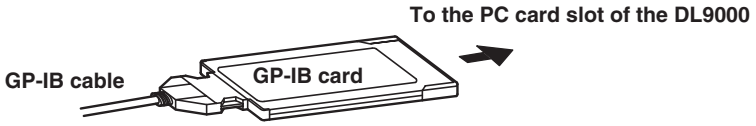
Use the NI PCMCIA-GPIB card by National Instruments.

GP-IB Cable

Use the GP-IB cable that comes with the GP-IB card.

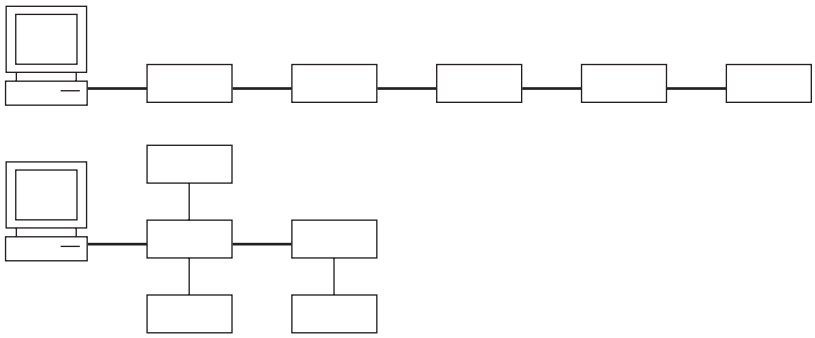
Connection Procedure

Insert the GP-IB card into the DL9000 PC card slot.



Precautions to Be Taken When Making Connections

- For the handling of the GP-IB card, see the manual that comes with the GP-IB card.
- Attach the GP-IB correctly with the front face up.
- Connect the GP-IB card to the DL9000 PC card slot first. Then, turn the DL9000 ON.
- Multiple cables can be used to connect multiple devices. However, no more than 15 devices including the controller can be connected on a single bus.
- When connecting multiple devices, each device must have its own unique address.
- Use a cable of length 2 m or less for connecting the devices.
- Make sure the total cable length does not exceed 20 m.
- When communicating, have at least two-thirds of the devices turned ON.
- When connecting multiple devices, connect them in a star or linear configuration (see the figure below). Do not wire them in a loop or parallel configuration.



CAUTION

When connecting or disconnecting communication cables, make sure to turn OFF the PC and the DL9000. Otherwise, erroneous operation or damage to the internal circuitry may result.

3.3 GP-IB Interface Functions

GP-IB Interface Functions

Listener Capability

- All of the information that you can set with the panel keys can be set through the GP-IB interface except for turning ON/OFF the power and setting the communication parameters.
- Receives commands from a controller requesting the output of setup information, waveform data, and other information.
- Also receives status report commands.

Talker Capability

Outputs setup information, waveform data, and other information.

Note

Listen-only, talk-only, and controller capabilities are not available on the DL9000.

Switching between Remote and Local Modes

When Switching from Local to Remote Mode

Receiving a REN (Remote Enable) message from the PC when the DL9000 is in the local mode causes the DL9000 to switch to the remote mode.

- REMOTE is displayed in the center of the upper section of the screen.
- All keys except the **CLEAR** key are disabled.
- Settings entered in local mode are retained even when the DL9000 switches to remote mode.

When Switching from Remote to Local Mode

Pressing **CLEAR** in remote mode puts the DL9000 in local mode. However, this act is invalid if the DL9000 has been set to Local Lockout mode (see page 3-6) by the controller.

- REMOTE indication in the center of the upper section of the screen disappears.
- Key operations are enabled.
- Settings entered in remote mode are retained even when the DL9000 switches to local mode.

Note

The GP-IB interface cannot be used simultaneously with other interfaces (USB and network interfaces).

3.4 GP-IB Interface Specifications

GP-IB Interface Specifications

Electrical and mechanical specifications:	Conforms to IEEE St'd 488-1978
Functional specifications:	See table below.
Protocol:	Conforms to IEEE St'd 488.2-1992
Code used:	ISO (ASCII) code
Mode:	Addressable mode
Address setting:	The address can be set in the range from 0 to 30 on the GP-IB setting screen that is played using the MISC menu.
Clear remote mode:	Remote mode can be cleared by pressing **CLEAR** except when the DL9000 has been set to Local Lockout mode by the controller.

Functional specifications

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), and no talk-only capability
Listener	L4	Basic listener capability, unlisten on MTA (My Talk Address), and no listen-only capability.
Service request	SR1	Full service request capability
Remote local	RL1	Full remote/local capability
Parallel polling	PP0	No parallel polling capability
Device clear	DC1	Full device clear capability
Device trigger	DT0	No device trigger capability
Controller	C0	No controller capability
Electrical characteristics	E1	Open collector

Data Rate

The response time when outputting waveform data is shown below as a reference.

Model: 701313

Controller: PC (Pentium4 3.4 GHz, USB 2.0), OS (Windows XP Professional SP1)

Language used: Visual C++

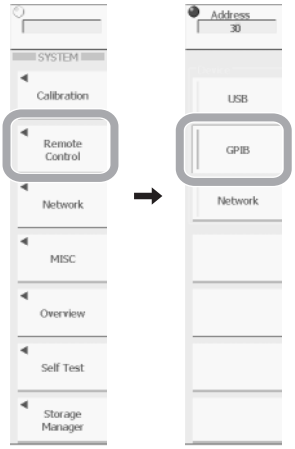
Number of Data Points	Word Data	ASCII Data
2s500	Approx. 16 ms	Approx. 0.390 s
125000	Approx. 344 ms	Approx. 19.453 s
1250000	Approx. 3172 ms	Approx. 194.516 s
2500000	Approx. 6282 ms	Approx. 389.047 s
6250000	Approx. 15641 ms	Approx. 971.985 s

3.5 Setting the DL9000 (GP-IB)

Procedure

Selecting the Communication Interface

1. Press **SYSTEM**.
2. Press the **Remote Control** soft key to display the Device menu.
3. Press the **GP-IB** soft key.



Note

Only the communication interface selected by Remote Control is enabled. The DL9000 does not accept commands that are transmitted to other unselected communication interfaces.

Setting the Address

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

Explanation

Enter the following settings when using a controller to set information that can be specified through key operation on the DL9000 or when outputting setting parameters or output waveform data to the controller.

Setting the Address

Set the address of the DL9000 within the following range for the addressable mode.

0 to 30

Each device that can be connected via GP-IB has a unique address within the GP-IB system. This address is used to distinguish the device from others. Therefore, when you connect the DL9000 to a PC, for example, make sure to assign a unique address to the DL9000.

Note

Do not change the address while the controller is communicating with the DL9000 or other devices over the GP-IB.

3.6 Responses to Interface Messages

Responses to Interface Messages

Responses to a Uni-Line Message

IFC (Interface Clear)

Clears the talker and listener functions. Stops output if data are being output.

REN (Remote Enable)

Switches between the remote and local modes.

IDY (Identify) is not supported.

Responses to a Multi-Line Message (Address Command)

GTL (Go To Local)

Switches to the local mode.

SDC (Selected Device Clear)

- Clears the program message (command) being received and the output queue (see page 6-5).
- *OPC and *OPC? commands in execution are void.
- The *WAI and COMMunicate:WAIT commands are immediately terminated.

PPC (Parallel Poll Configure), GET(Group Execute Trigger), and TCT (Take Control) are not supported.

Responses to a Multi-Line Message (Universal Command)

LLO (Local Lockout)

Disables **CLEAR** on the front panel to prohibit switching to the local mode.

DCL (Device Clear)

Same operation as the SDC message.

SPE (Serial Poll Enable)

Sets the talker function on all devices on the bus to serial polling mode. The controller polls the devices in order.

SPD (Serial Poll Disable)

Clears the serial polling mode of the talker function on all devices on the bus.

PPU (Parallel Poll Unconfigure) is not supported.

What Are Interface Messages

Interface messages are also referred to as interface commands or bus commands. They are commands that are issued by the controller. They are classified as follows:

Uni-Line Messages

A single control line is used to transmit uni-line messages. The following three types are available.

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

Multi-Line Messages

Eight data lines are used to transmit multi-line messages. The messages are classified as follows:

Address Commands

These commands are valid when the instrument is designated as a listener or as a talker. The following five types are available.

Commands that are valid on an instrument that is designated as a listener

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

Commands that are valid on an instrument that is designated as a talker

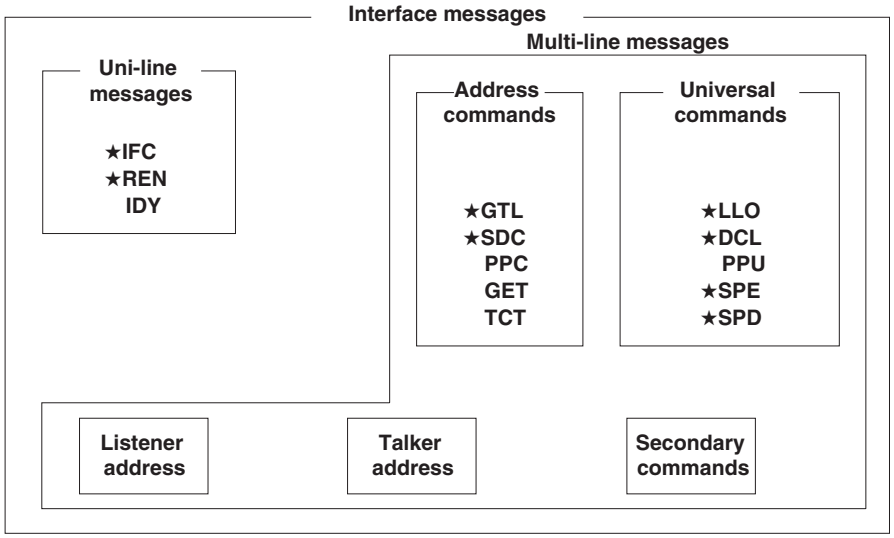
- TCT (Take Control)

Universal commands

These commands are valid on all instruments regardless of the listener and talker designations. The following five types are available.

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

In addition, listener address, talker address, and secondary commands are also considered interface messages.



Interface messages that DL9000 supports are indicated with ★marks.

Note

The Differences between SDC and DCL

In multi-line messages, SDC messages are those that require talker or listener designation and DCL messages are those that do not require the designation. Therefore, SDC messages are directed at a particular instrument while DCL messages are directed at all instruments on the bus.

Chapter 4 Before Programming

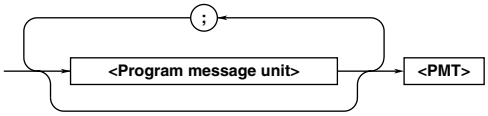
4.1 Messages

Messages

Messages are used to exchange information between the controller and the instrument. Messages that are sent from the controller to the instrument are called program messages and messages that are sent back from the instrument to the controller are called response messages.
If a program message contains a message unit that requests a response (a query), the instrument returns a response message upon receiving the program message. A single response message is always returned in response to a single program message.

Program Messages

The program message format is shown below.



<Program Message Unit>

A program message consists of zero or more program message units; each unit corresponds to one command. The instrument executes the received commands in order.
Each program message unit is separated by a semicolon (`;`).
For details regarding the format of the program message unit, see the next section.

Example

```
:ACQuire:MODE NORMal;HRMode 1<PMT>
```

Unit Unit

<PMT>

PMT is a program message terminator. The following three types are available.

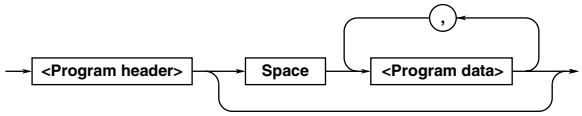
`NL` (New Line): Same as LF (Line Feed). ASCII code "0AH"

`^EOM`: The END message as defined by USBTMC (The data byte that is sent simultaneously with the END message is the last data of the program message.)

`NL^EOM`: NL with an END message added (NL is not included in the program message.)

- **Program Message Unit Format**

The program message unit format is shown below.



<Program Header>

The program header indicates the command type. For details, see page 4-3.

<Program Data>

If certain conditions are required in executing a command, program data is added. A space (ASCII code "20H") separates the program data from the header. If there are multiple sets of program data, they are separated by commas (`,`).
For details, see page 4-5.

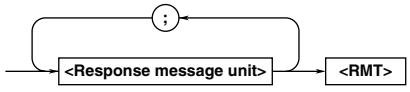
Example

```
:ACQuire:MODE NORMal<PMT>
```

Header Data

Response Messages

The response message format is shown below.



<Response Message Unit>

A response message consists of one or more response message units; each response message unit corresponds to one response.
Response message units are separated by a semicolon (`;`).
For details regarding the format of the response message unit, see the next section.

Example

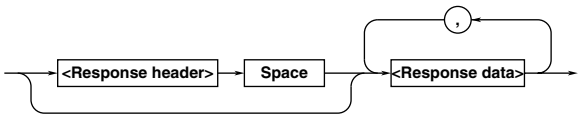
```
:ACQUIRE:MODE NORMAL;HRMode 1<RMT>
```

Unit Unit

<RMT>

A response message terminator. It is `NL^EOM`.

- **Response Message Unit Format**
  The response message unit format is shown below.



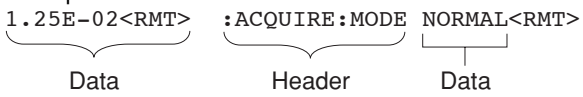
<Response Header>

A response header sometimes precedes the response data. A space separates the data from the header. For details, see page 4-4.

<Response Data>

Response data contains the content of the response. If there are multiple sets of response data, they are separated by commas (,). For details, see page 4-5.

Example



If there are multiple queries in a program message, responses are made in the same order as the queries. In most cases, a single query returns a single response message unit, but there are a few queries that return multiple units. The first response message unit always corresponds to the first query, but the n^{th} response unit may not necessarily correspond to the n^{th} query. Therefore, if you want to make sure that every response is retrieved, divide the program messages into individual messages.

Precautions to Be Taken when Transferring Messages

- If a program message that does not contain a query is sent, the next program message can be sent at any time.
- If a program message that contains a query is sent, a response message must be received before the next program message can be sent. If the next program message is sent before the response message is received in its entirety, an error occurs. The response message that was not received is discarded.
- If the controller tries to receive a response message when there is none, an error occurs. If the controller tries to receive a response message before the transmission of the program message is complete, an error occurs.
- If a program message containing multiple message units is sent, and the message contains incomplete units, the instrument attempts to execute the ones that are believed to be complete. However, these attempts may not always be successful. In addition, if the message contains queries, the responses may not be returned.

Deadlock

The instrument can store in its buffer program and response messages of length 1024 bytes or more (The number of available bytes varies depending on the operating conditions). When both the transmit and receive buffers become full at the same time, the instrument can no longer continue to operate. This state is called a deadlock. In this case, operation can be resumed by discarding the program message. Deadlock will not occur if the program message (including the <PMT>) is kept below 1024 bytes. Furthermore, deadlock never occurs if a program message does not contain a query.

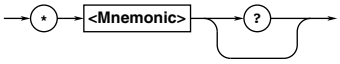
4.2 Commands

Commands

There are three types of commands (program headers) that are sent from the controller to the instrument. They differ in their program header formats.

Common Command Header

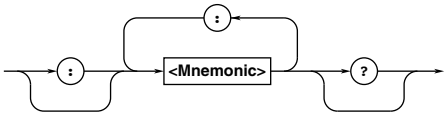
Commands that are defined in the USBTMC-USB488 are called common commands. The header format of a common command is shown below. An asterisk (*) is always placed in the beginning of a command.



Common command example: `*CLS`

Compound Header

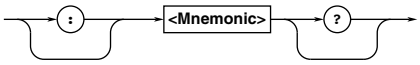
Dedicated commands used by the instrument are classified and arranged in a hierarchy according to their functions. The format of a compound header is shown below. A colon (:) must be used to specify a lower hierarchy.



Compound header example: `:ACQuire:MODE`

Simple Header

These commands are functionally independent and do not have a hierarchy. The format of a simple header is shown below.



Simple header example: `: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 that have common compound headers arranged in a hierarchy. A command group may contain sub-groups.

Example Group of commands related to acquisition

```
:ACQuire:AVERage:COUNt
:ACQuire:MODE
:ACQuire:AVERage:EWEight
:ACQuire:REPetitive
:ACQuire:RLENgth
:ACQuire:INTERLeave
```

- **When Concatenating Commands of the Same Group**

  The instrument stores the hierarchical level of the command that is currently being executed, and performs analysis on the assumption that the next command sent will also belong to the same level. Therefore, common header sections can be omitted for commands belonging to the same group.

  Example
  ```
  :ACQuire:MODE NORMal;
  INTERLeave 1<PMT>
  ```

- **When Concatenating Commands of Different Groups**

  If the following command does not belong to the same group, a colon (`:`) is placed in front of the header (cannot be omitted).

  Example
  ```
  :ACQuire:MODE NORMal;:DISPlay:
  FORMat SINGle<PMT>
  ```

- **When Concatenating Simple Headers**

  If a simple header follows another command, a colon (`:`) is placed in front of the simple header (cannot be omitted).

  Example
  ```
  :ACQuire:MODE
  NORMal;:STARt<PMT>
  ```

- **When Concatenating Common Commands**

  Common commands that are defined in the USBTMC-USB488 are independent of hierarchy. Colons (`:`) are not needed before a common command.

  Example
  ```
  :ACQuire:MODE NORMal;*CLS;
  INTERLeave 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 specified for each command even when commands belonging to the same command group are being concatenated.

  Example
  ```
  :ACQuire:MODE NORMal<PMT>:
  ACQuire:INTERLeave 1<PMT>
  ```

Upper-Level Query

An upper-level query is a query in which a question mark (?) is appended to the highest level command of a group. Execution of an upper-level query allows all settings that can be specified in the group to be received at once. Some query groups which are comprised of more than three hierarchical levels can output all the lower level settings.

Example
```
:ACQUIRE? ->
:ACQUIRE:AVERAGE:COUNT 2;
EWEIGHT 16;:ACQUIRE:HRMODE 0;
INTERLEAVE 0;INTERPOLATE 1;
MODE NORMAL;REPETITIVE 0;
RLENGTH 12500
```

The response to an upper-level query can be transmitted as a program message back to the instrument. In this way, the settings that existed when the upper-level query was made can be restored. However, some upper-level queries do not return setup information that is not currently in use. It is important to remember that not all the group's information is necessarily returned as part of a response.

Header Interpretation Rules

The instrument interprets the header that is received according to the rules below.

- Mnemonics are not case sensitive.
  Example "CURSor" can also be written as "cursor" or "Cursor."
- The lower-case section of the header can be omitted.
  Example "CURSor" can also be written as "CURSO" or "CURS."
- The question mark (?) at the end of a header indicates that it is a query. The question mark (?) cannot be omitted.
  Example The shortest abbreviation for CURSor? is CURS?.
- If the <x> (value) at the end of a mnemonic is omitted, it is interpreted as a 1.
  Example If "CHANnel<x>" is written as "CHAN," it means "CHANnel1."
- **The section enclosed by braces ([]) can be omitted.**
  Example HISTory[:CURRent]:MODE can also be written as HIST:MODE.
  However, the last section enclosed by braces ([]) cannot be omitted in an upper-level query.
  Example "HISTory?" and "HISTory:CURRent?" are different queries.

4.3 Response

Response

When the controller sends a message unit that has a question mark (?) in its program header (query), the instrument returns a response message to the query. A response message is returned in one of the following two forms.

- Response Consisting of a Header and Data
  If the response can be used as a program message without any change, it is returned with a command header attached.
  Example
  ```
  :ACQUire:MODE?<PMT> ->
  :ACQUire:MODE NORMAL<RMT>
  ```

- Response Consisting of Data Only
  If the response cannot be used as a program message unless changes are made to it (query-only command), only the data section is returned. However, there are query-only commands that return responses with the header attached.
  Example
  ```
  :MEASure:TRACe1:AREA1:PTOPeak:
  VALue?<PMT> -> 10.0E+00<RMT>
  ```

- **When You Wish to Return a Response without a Header**
  Responses that return both header and data can be set so that only the data section is returned. The "COMMunicate:HEADer" command is used to do this.

- **Abbreviated Form**
  Normally, the lower-case section 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 sections enclosed by braces ([]) are also omitted in the abbreviated form.

4.4 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	Meaning
<Decimal>	A value expressed as a decimal number (Example: Probe attenuation of CH1 -> CHANnel1:PROBe 100)
<Voltage><Time> <Frequency> <Current>	A physical value (Example: Time axis range -> TIMebase:TDIV 1US)
<Register>	Register value expressed as binary, octal, decimal or hexadecimal. (Example: Extended event register value -> STATUS:EESE #HFE)
<Character Data>	Predefined character string (mnemonic). Can be selected from { }. (Example: Select the input coupling of CH1 -> CHANnel1:COUPling {AC|DC|DC50|GND})
<Boolean>	Indicates ON and OFF. Set using ON, OFF or a value (Example: Turn ON the CH1 display -> CHANnel1:DISPlay ON)
<String data>	An arbitrary character string (Example: Comment to a screen data output -> MATH1:UNIT:USERdefine "VOLT")
<Filename>	Indicates a file name. (Example: Save file name -> 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 as specified in the ANSI X3.42-1975.

Symbol	Meaning	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.			

- The instrument can receive decimal values that are sent from the controller in any of the forms, <NR1> to <NR3>. This is represented by <NRf>.
- For response messages that the instrument returns to the controller, the form (<NR1> to <NR3> to be used) is determined by the query. The same form is used regardless of the size of the value.
- For the <NR3> format, the "+" sign after the "E" can be omitted. However, the "-" sign cannot be omitted.
- If a value outside the setting range is entered, the value is normalized so that it is just inside the range.
- If a value has more significant digits than the available resolution, the value is rounded.

<Voltage>, <Time>, <Frequency>, and <Current>

<Voltage>, <Time>, <Frequency>, and <Current> indicate decimal values that have physical significance. <Multiplier> or <Unit> can be attached to the <NRf> form that was described earlier. It is expressed in one 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 indicated below.

Symbol	Word	Multiplier
EX	Exa	10^{18}
PE	Peta	10^{15}
T	Tera	10^{12}
G	Giga	10^{9}
MA	Mega	10^{6}
K	Kilo	10^{3}
M	Milli	10^{-3}
U	Micro	10^{-6}
N	Nano	10^{-9}
P	Pico	10^{-12}
F	Femto	10^{-15}
A	Ato	10^{-18}

- **<Unit>**
  <Units> that can be used are indicated below.

Symbol	Word	Meaning
V	Volt	Voltage
S	Second	Time
HZ	Hertz	Frequency
MHZ	Megahertz	Frequency
A	Ampere	Current

- <Multiplier> and <Unit> are not case sensitive.
- "U" is used to indicate micro "μ".
- "MA" is used for Mega to distinguish it from Milli. The only exception is Megahertz which is expressed as "MHZ." Therefore, the "M (Milli)" multiplier cannot be used for frequencies.
- If both <Multiplier> and <Unit> are omitted, the default unit is used.
- Response messages are always expressed in the <NR3> form. Response messages are returned using the default unit without the <Multiplier> or <Unit>.

<Register>

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

Form	Example
<NRf>	1
#H<Hexadecimal value made up of the digits 0 to 9 and A to F>	#H0F
#Q<Octal value made up of the digits 0 to 7>	#Q777
#B<Binary value made up of the digits 0 and 1>	#B001100

- <Register> is not case sensitive.
- Response messages are always expressed as <NR1>.

<Character Data>

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

Form	Example
{AC|DC|DC50|GND}	AC

- As with the header, the "COMMunicate:VERBose" command can be used to select whether to return the response in the full form or in the abbreviated form.
- The "COMMunicate:HEADer" setting does not affect the character data.

<Boolean>

<Boolean> is data that indicates ON or OFF. It is expressed in one of the following forms.

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

- When <Boolean> is expressed in the <NRf> form, "OFF" is selected if the rounded integer value is 0, and ON for all other cases.
- A response message is always returned with a 1 if the value is ON and 0 if the value is OFF.

<String data>

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

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

- If a character string contains a double quotation mark ('), the double quotation mark is replaced by two double quotation marks (""). This rule also applies to a single quotation mark within a character string.
- A response message is always enclosed in double quotation marks (").
- <String data> is an arbitrary character string. Therefore the 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 is detected if a quotation mark is omitted.

<Filename>

<Filename> is data that indicates a file name. It is expressed in one of the following forms.

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

- <NRf> is rounded to an 8-digit integer and converted to ASCII code. The result is the file name (example: 1 becomes "00000001"). Negative values are not allowed.
- Response messages are always returned in the <String data> form.
- For <Character data>, the first 12 characters become the file name.
- For <String data>, the first 259 characters become the file name.
- For a description of the number of characters of the <String data> file name, see the *DL9000 User's Manual*.

<Block data>

<Block data> is arbitrary 8-bit data. It is only used in response messages on the DL9000. Below is the syntax.

Form	Example
#N<N-digit decimal number><Data byte sequence>	#800000010ABCDEFGHIJ

- #N
  Indicates that the data is <Block data>. "N" indicates the number of succeeding data bytes (digits) in ASCII code characters.
- <N-digit decimal number>
  Indicates the number of bytes of data (example: 00000010 = 10 bytes).
- <data byte sequence>
  Expresses 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.

4.5 Synchronization with the Controller

Overlap Commands and Sequential Commands

There are two types of commands, overlap commands and sequential commands. In the case of overlap commands, the execution of the next command may start before the execution of the previous command is completed.

For example, if the next program message is transmitted when specifying the V/div value and querying the result, the response always returns the most recent setting (5 V in this case).

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

This is because the next command is forced to wait until the processing of "CHANnel1:VDIV" itself is completed. This type of command is called a sequential command.

On the contrary, let us assume that you send the next program message when you wish to load a file and query the V/div value of the result.

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

In this case, "CHANnel1:VDIV?" is executed before the loading of the file is completed, and the V/div value that is returned is the value before the file is loaded.

The act of executing the next command before the processing of itself is completed such as with "FILE:LOAD:SETup:EXECute "CASE1"" is called an overlap operation. A command that operates in this way is called an overlap command.

In such case, the overlap operation can be prevented by using the methods below.

Synchronizing with Overlap Commands

• Using the *WAI Command

The *WAI command holds the subsequent commands until the overlap command is completed.

Example
```
:COMMunicate:OPSE #H0040;:
FILE:LOAD:SETup:
EXECute "CASE1";*WAI;:
CHANnel1:VDIV?<PMT>
```

"COMMunicate:OPSE" is a command used to select the "*WAI" target. Here, media access is specified.

Because "*WAI" is executed immediately before "CHANnel1:VDIV?," "CHANnel1:VDIV?" is not executed until the file loading is complete.

• Using the COMMunicate:OVERlap command

The COMMunicate:OVERlap command enables (or disables) overlap operation.

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

"COMMunicate:OVERlap #HFFBF" enables overlap operation on commands other than media access. Because the overlap operation of file loading is disabled, "FILE:LOAD:SETup:EXECute "CASE1"" operates in the same way as a sequential command. Therefore, CHANnel1:VDIV? is not executed until the file loading is complete.

• Using the *OPC Command

The *OPC command sets the OPC bit, bit 0 of the standard event register (see page 6-3), to 1 when the overlap operation is completed.

Example
```
:COMMunicate:OPSE #H0040;
*ESE 1;
*ESR?;*SRE 32;:FILE:LOAD:SETup:
EXECute "CASE1";*OPC<PMT>
(Read the response to *ESR?)
(Wait for a service request)
:CHANnel1:VDIV?<PMT>
```

"COMMunicate:OPSE" is a command used to select the "*OPC" target. Here, media access is specified.

"*ESE 1" and "*SRE 32" indicate that a service request is generated only when the OPC bit is 1.

"*ESR?" clears the standard event register.

In the example above, "CHANnel1:VDIV?" is not executed until a service request is generated.

- **Using the *OPC? Query**
  The *OPC? query generates a response when an overlap operation is completed.
  Example :COMMunicate:OPSE #H0040;:FILE:
  LOAD:SETup:EXECute "CASE1";
  *OPC?<PMT>
  (Read the response to *OPC?)
  :CHANnel1:VDIV?<PMT>
  "COMMunicate:OPSE" is a command used to select the "*OPC?" target. Here, media access is specified.
  Because "*OPC?" does not generate a response until the overlap operation is completed, the loading of the file will have been completed by the time the response to "*OPC?" is read.

Note

Most commands are sequential commands. Overlap commands are indicated as overlap commands in chapter 6. All other commands are sequential commands.

Achieving Synchronization without Using Overlap Commands

Even for sequential commands, synchronization is sometimes required for non communication-related reasons such as a trigger occurrence.
For example, if the next program message is transmitted to make an inquiry about the waveform data which has been acquired with the trigger mode set to single, the WAVeform:SEND? command may be executed regardless of whether the acquisition has been completed or not and may result in command execution error.
TRIGger:MODE SINGle;:STARt;:WAVeform:SEND?<PMT>
In this case, the following method must be used to synchronize with the end of the acquisition.

- **Using the STATus:CONDition? Query**
  The "STATus:CONDition?" query is used to query the contents of the condition register (page 6-4). Whether waveforms are being retrieved can be determined by reading bit 0 of the condition register. If bit 0 of the condition register is "1," waveforms are being retrieved. Otherwise, it is stopped.
  Example TRIGger:MODE SINGle;:STARt<PMT>
  :STATus:CONDition?<PMT>
  (Read the response. If bit 0 is 1, repeat this command until it becomes 1.)
  :WAVeform:SEND?<PMT>
  The WAVeform:SEND? command will not be executed until bit 0 of the condition register is set to "0."

- **Using the Extended Event Register**
  The changes in the condition register can be reflected in the extended event register (page 6-4).
  Example :STATus:FILTer1 FALL;:
  STATus:EESE 1;EESR?;*SRE 8;:
  TRIGger:MODE SINGle;:STARt<PMT>
  (Read the response to STATus:EESR?)
  (Wait for a service request)
  :WAVeform:SEND?<PMT>
  The "STATus:FILTer1 FALL" command sets the transition filter so that bit 0 (FILTer1) of the extended event register is set to 1 when bit 0 of the condition register changes from 1 to 0.

  The ":STATus:EESE 1" command is used to reflect only bit 0 of the extended event register to the status byte.
  The "STATus:EESR?" command is used to clear the extended event register.
  The "*SRE 8" command is used to generate a service request solely on the cause of the extended event register.
  The "WAVeform:SEND?" command is not 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<PMT>
  (Read the response to STATus:EESR?)
  :COMMunicate:WAIT 1;:WAVeform:
  SEND?<PMT>
  For a description of "STATus:FILTer1 FALL" and "STATus:
  EESR?" see the previous section regarding the extended event register.
  The "COMMunicate:WAIT 1" command indicates that the program will wait for bit 0 of the extended event register to be set to "1."
  The WAVeform:SEND? command will not be executed until bit 0 of the extended event register is set to "1."

Chapter 5 Commands

5.1 A List of Commands

Command	Function	Page
:ANALysis:AHIStogram<x>:MEASure:CURSor:HLINkage	Turns ON/OFF the horizontal cursor link of the accumulated histogram or queries the current setting.	5-38
:ANALysis:AHIStogram<x>:MEASure:CURSor:HPOSition<x>	Sets the horizontal cursor position or queries the current setting.	5-38
:ANALysis:AHIStogram<x>:MEASure:CURSor:VLINkage	Turns ON/OFF the vertical cursor link of the accumulated histogram or queries the current setting.	5-38
:ANALysis:AHIStogram<x>:MEASure:CURSor:VPOSition<x>	Sets the vertical cursor position or queries the current setting.	5-39
:ANALysis:AHIStogram<x>:MEASure:MODE	Sets the automated measurement mode or queries the current setting.	5-39
:ANALysis:AHIStogram<x>:MEASure:PARameter?	Queries all settings related to the automated measurement of waveform parameters.	5-39
:ANALysis:AHIStogram<x>:MEASure:PARameter:AREA<x>?	Queries all settings related to the area.	5-40
:ANALysis:AHIStogram<x>:MEASure:PARameter:AREA<x>:ALL	Turns ON/OFF all waveform parameters.	5-40
:ANALysis:AHIStogram<x>:MEASure:PARameter:AREA<x>:<Parameter>?	Queries all settings related to the waveform parameter.	5-40
:ANALysis:AHIStogram<x>:MEASure:PARameter:AREA<x>:<Parameter>:STATe	Turns ON/OFF the waveform parameter or queries the current setting.	5-40
:ANALysis:AHIStogram<x>:MEASure:PARameter:AREA<x>:<Parameter>:VALue?	Queries the automated measured value of the waveform parameter.	5-41
:ANALysis:AHIStogram<x>:MEASure:PARameter:CALCulation?	Queries all settings related to the calculation items of waveform parameters.	5-41
:ANALysis:AHIStogram<x>:MEASure:PARameter:CALCulation:ALL	Turns ON/OFF all calculation items of the waveform parameters.	5-41
:ANALysis:AHIStogram<x>:MEASure:PARameter:CALCulation:DEFine<x>	Sets the equation of the calculation items of the waveform parameter or queries the current setting.	5-41
:ANALysis:AHIStogram<x>:MEASure:PARameter:CALCulation:STATe<x>	Turns ON/OFF the calculation items of the waveform parameter or queries the current setting.	5-41
:ANALysis:AHIStogram<x>:MEASure:PARameter:CALCulation:VALue<x>?	Queries the automated measured value of the calculation items of the waveform parameter.	5-41
:ANALysis:AHIStogram<x>:MEASure:PARameter:HRANge<x>	Sets the horizontal range of the waveform parameter or queries the current setting.	5-42
:ANALysis:AHIStogram<x>:MEASure:PARameter:VRANge<x>	Sets the vertical range of the waveform parameter or queries the current setting.	5-42
:ANALysis:AHIStogram<x>:MODE	Sets the accumulated histogram mode or queries the current setting.	5-42
:ANALysis:AHIStogram<x>:TRACe	Sets the source trace of the accumulated histogram or queries the current setting.	5-42
:ANALysis:AHIStogram<x>:VERTical	Sets the vertical range of the accumulated histogram or queries the current setting.	5-42
:ANALysis:AHIStogram<x>:WINDow	Sets the measurement target window of the accumulated histogram or queries the current setting.	5-42
:ANALysis:DISPlay<x>	Turns ON/OFF the analysis function display or queries the current setting.	5-42
:ANALysis:FFT<x>?	Queries all settings related to the FFT computation function.	5-43
:ANALysis:FFT<x>:HORizontal?	Queries all settings related the horizontal axis of the FFT computation.	5-43

Command	Function	Page
:ANALysis:FFT<x>:HORizontal:CSPan?	Queries all settings related to the center and span of the horizontal axis of the FFT computation.	5-43
:ANALysis:FFT<x>:HORizontal:CSPan:CENTer	Sets the horizontal center of the FFT computation or queries the current setting.	5-43
:ANALysis:FFT<x>:HORizontal:CSPan:SPAN	Sets the horizontal span of the FFT computation or queries the current setting.	5-43
:ANALysis:FFT<x>:HORizontal:LRIGht?	Queries all settings related the left and right edges of the horizontal axis of the FFT computation.	5-43
:ANALysis:FFT<x>:HORizontal:LRIGht:RANGe	Sets the range of the horizontal left and right edges of the FFT computation or queries the current setting.	5-44
:ANALysis:FFT<x>:HORizontal:MODE	Sets the horizontal mode of the FFT computation or queries the current setting.	5-44
:ANALysis:FFT<x>:IPARt	Sets the source trace of the imaginary part of the FFT computation or queries the current setting.	5-44
:ANALysis:FFT<x>:LENGth	Sets the number of FFT points or queries the current setting.	5-44
:ANALysis:FFT<x>:MAXHold	Turns ON/OFF the maximum value hold function of the FFT computation or queries the current setting.	5-44
:ANALysis:FFT<x>:MEASure?	Queries all settings related to the automated measurement of the FFT computation.	5-44
:ANALysis:FFT<x>:MEASure:MARKer?	Queries all settings related to the marker cursor measurement of the FFT computation.	5-45
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]?	Queries all settings related to basic items of the marker cursor of the FFT computation.	5-45
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:ALL	Turns ON/OFF all basic items of the marker cursor of the FFT computation.	5-45
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:DFRequency?	Queries all settings related to the frequency value between marker cursors of the FFT computation.	5-45
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:DFRequency:STATe	Turns ON/OFF the frequency value between marker cursors of the FFT computation or queries the current setting.	5-45
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:DFRequency:VALue?	Queries the frequency value between marker cursors of the FFT computation.	5-45
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:DV?	Queries all settings related to the power value between marker cursors of the FFT computation.	5-46
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:DV:STATe	Turns ON/OFF the power value between marker cursors of the FFT computation or queries the current setting.	5-46
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:DV:VALue?	Queries the power value between marker cursors of the FFT computation.	5-46
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:FREQuency<x>?	Queries all settings related to the frequency value of the marker cursor of the FFT computation.	5-46

Command	Function	Page
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:FREQuency<x>:STATe	Turns ON/OFF the frequency value of the marker cursor of the FFT computation or queries the current setting.	5-46
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:FREQuency<x>:VALue?	Queries the frequency value of the marker cursor of the FFT computation.	5-46
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:POSition<x> {<NRf>}	Sets the marker cursor position of the FFT computation or queries the current setting.	5-46
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:V<x>?	Queries all settings related to the power value of the marker cursor of the FFT computation.	5-47
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:V<x>:STATe	Turns ON/OFF the power value of the marker cursor of the FFT computation or queries the current setting.	5-47
:ANALysis:FFT<x>:MEASure:MARKer[:BASic]:V<x>:VALue?	Queries the power value of the marker cursor of the FFT computation.	5-47
:ANALysis:FFT<x>:MEASure:MARKer:CALCulation?	Queries all settings related to calculation items of the marker cursor of the FFT computation.	5-47
:ANALysis:FFT<x>:MEASure:MARKer:CALCulation:ALL	Turns ON/OFF all calculation items of the marker cursor of the FFT computation.	5-47
:ANALysis:FFT<x>:MEASure:MARKer:CALCulation:DEFine<x>	Sets the equation of the calculation items of the marker cursor of the FFT computation or queries the current setting.	5-47
:ANALysis:FFT<x>:MEASure:MARKer:CALCulation:STATe<x>	Turns ON/OFF the calculation items of the marker cursor of the FFT computation or queries the current setting.	5-47
:ANALysis:FFT<x>:MEASure:MARKer:CALCulation:VALue<x>?	Queries the measured value of the calculation items of the marker cursor of the FFT computation.	5-48
:ANALysis:FFT<x>:MEASure:MODE	Sets the automated measurement mode of the FFT computation or queries the current setting.	5-48
:ANALysis:FFT<x>:MEASure:PEAK?	Queries all settings related to the peak value measurement of the FFT computation.	5-48
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]?	Queries all settings related to basic items of the peak value of the FFT computation.	5-48
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:ALL	Turns ON/OFF all basic items of the peak value of the FFT computation.	5-48
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:DFRequency?	Queries all settings related to the frequency value between peak values of the FFT computation.	5-48
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:DFRequency:STATe	Turns ON/OFF the frequency value between peak values of the FFT computation or queries the current setting.	5-49
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:DFRequency:VALue?	Queries the frequency value between peak values of the FFT computation.	5-49
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:DV?	Queries all settings related to the power value between peak values of the FFT computation.	5-49

Command	Function	Page
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:DV:STATe	Turns ON/OFF the power value between peak values of the FFT computation or queries the current setting.	5-49
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:DV:VALue?	Queries the power value between peak values of the FFT computation.	5-49
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:FREQuency<x>?	Queries all settings related to the peak frequency value of the FFT computation.	5-49
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:FREQuency<x>:STATe	Turns ON/OFF the peak frequency value of the FFT computation or queries the current setting.	5-49
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:FREQuency<x>:VALue?	Queries the peak frequency value of the FFT computation.	5-49
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:RANGe<x>	Sets the measurement range of the peak value of the FFT computation or queries the current setting.	5-50
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:V<x>?	Queries all settings related to the peak value of the FFT computation.	5-50
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:V<x>:STATe	Turns ON/OFF the peak value of the FFT computation or queries the current setting.	5-50
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:V<x>:VALue?	Queries the peak value of the FFT computation.	5-50
:ANALysis:FFT<x>:MEASure:PEAK:CALCulation?	Queries all settings related to calculation items of the FFT computation.	5-50
:ANALysis:FFT<x>:MEASure:PEAK:CALCulation:ALL	Turns ON/OFF all calculation items of the FFT computation.	5-50
:ANALysis:FFT<x>:MEASure:PEAK:CALCulation:DEFine<x>	Sets the equation of the calculation item of the FFT computation or queries the current setting.	5-50
:ANALysis:FFT<x>:MEASure:PEAK:CALCulation:STATe<x>	Turns ON/OFF the calculation items of the FFT computation or queries the current setting.	5-51
:ANALysis:FFT<x>:MEASure:PEAK:CALCulation:VALue<x>?	Queries the measured value of the calculation item of the FFT computation.	5-51
:ANALysis:FFT<x>:RANGe	Sets the measurement source window used in the FFT computation or queries the current setting.	5-51
:ANALysis:FFT<x>:RPARt	Sets the source trace of the real part of the FFT computation or queries the current setting.	5-51
:ANALysis:FFT<x>:RPOSition	Sets the center point of magnification of the vertical axis of the FFT computation or queries the current setting.	5-51
:ANALysis:FFT<x>:VERTical?	Queries all settings related the vertical axis of the FFT computation.	5-51
:ANALysis:FFT<x>:VERTical:LEVel	Sets the display position of the vertical axis of the FFT computation or queries the current setting.	5-51
:ANALysis:FFT<x>:VERTical:MODE	Sets the vertical axis mode of the FFT computation or queries the current setting.	5-51
:ANALysis:FFT<x>:VERTical:SENSitivity	Sets the vertical sensitivity of the FFT computation or queries the current setting.	5-52
:ANALysis:FFT<x>:WINDow	Sets the window function or queries the current setting.	5-52
:ANALysis:TYPE<x>	Sets the analysis function type or queries the current setting.	5-52
:ANALysis:VTDisplay	Turns ON/OFF the VT waveform display or queries the current setting.	5-52
:ANALysis:WAIT<x>?	Waits for the analysis to complete with a timeout.	5-52
:ANALysis:WPARameter<x>?	Queries all settings related to the waveform parameter measurement function.	5-52

Command	Function	Page
:ANALysis:WPARameter<x>:CALCulation	Sets the calculation items or queries the current setting.	5-53
:ANALysis:WPARameter<x>:HISTogram?	Queries all settings related to the histogram display.	5-53
:ANALysis:WPARameter<x>:HISTogram:MEASure?	Queries all settings related to the automated measurement of the histogram display.	5-53
:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor?	Queries all settings related to cursor measurements on the histogram.	5-53
:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:ALL	Turn ON/OFF all histogram cursors.	5-54
:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:C<x>?	Queries all settings related to the cursor measurement on the histogram.	5-54
:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:C<x>:POSition	Sets the cursor position on the histogram or queries the current setting.	5-54
:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:C<x>:STATe	Turns ON/OFF the cursor on the histogram or queries the current setting.	5-54
:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:C<x>:VALue?	Queries the measured value of the cursor on the histogram.	5-54
:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:DC?	Queries all settings related to the measurement between cursors on the histogram.	5-54
:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:DC:STATe	Turns ON/OFF the measurement between cursors on the histogram or queries the current setting.	5-55
:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:DC:VALue?	Queries the measured value between cursors on the histogram.	5-55
:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:LINKage	Turns ON/OFF the cursor link on the histogram or queries the current setting.	5-55
:ANALysis:WPARameter<x>:HISTogram:MEASure:MODE	Sets the automated measurement mode of the histogram display or queries the current setting.	5-55
:ANALysis:WPARameter<x>:HISTogram:MEASure:PARameter?	Queries all settings related to the automated measurement of histogram parameters.	5-55
:ANALysis:WPARameter<x>:HISTogram:MEASure:PARameter:ALL	Turn ON/OFF all histogram parameters.	5-55
:ANALysis:WPARameter<x>:HISTogram:MEASure:PARameter:<Parameter>?	Queries all settings related to the histogram parameter.	5-56
:ANALysis:WPARameter<x>:HISTogram:MEASure:PARameter:<Parameter>:STATe	Turns ON/OFF the histogram parameter or queries the current setting.	5-56
:ANALysis:WPARameter<x>:HISTogram:MEASure:PARameter:<Parameter>:VALue?	Queries the measured value of the histogram parameter.	5-56
:ANALysis:WPARameter<x>:LIST?	Queries all settings related to the list display.	5-56
:ANALysis:WPARameter<x>:LIST:ITEM?	Queries list display items.	5-56
:ANALysis:WPARameter<x>:LIST:SCRoll	Sets the scroll direction of the list display or queries the current setting.	5-56
:ANALysis:WPARameter<x>:LIST:VALue?	Queries the automated measured value of the list display number of the waveform parameter measurement.	5-56
:ANALysis:WPARameter<x>:MODE	Sets the mode of the waveform parameter measurement or queries the current setting.	5-57
:ANALysis:WPARameter<x>:TRACe<x>?	Queries all settings related to the trace of the waveform parameter measurement.	5-57
:ANALysis:WPARameter<x>:TRACe<x>:AREA<x>?	Queries all settings related to the area of the waveform parameter measurement.	5-57

Command	Function	Page
:ANALysis:WPARameter<x>:TRACe<x>:AREA<x>:TYPE		
	Sets the waveform parameter of the waveform parameter measurement or queries the current setting.	5-57
:ANALysis:WPARameter<x>:TRENd?	Queries all settings related to the trend display of the waveform parameter measurement.	5-57
:ANALysis:WPARameter<x>:TRENd:ASCale[:EXECute]		
	Executes the auto scaling of the trend display of the waveform parameter measurement.	5-58
:ANALysis:WPARameter<x>:TRENd:HSPan	Sets the horizontal span of the trend display of the waveform parameter measurement or queries the current setting.	5-58
:ANALysis:WPARameter<x>:TRENd:MEASure?		
	Queries all settings related to the automated measurement of the trend display of the waveform parameter measurement.	5-58
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor?		
	Queries all settings related to the cursor measurement of the trend of the waveform parameter measurement.	5-58
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:ALL		
	Turns ON/OFF all cursors of the trend of the waveform parameter measurement.	5-58
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:C<x>?		
	Queries all settings related to the measured value of the cursor of the trend of the waveform parameter measurement.	5-58
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:C<x>:POSition		
	Sets the cursor position of the trend of the waveform parameter measurement or queries the current setting.	5-58
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:C<x>:STATe		
	Turns ON/OFF the cursor of the trend of the waveform parameter measurement or queries the current setting.	5-59
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:C<x>:VALue?		
	Queries the measured value of the cursor of the trend of the waveform parameter measurement.	5-59
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:DC?		
	Queries all settings related to the measurement between cursors on the trend.	5-59
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:DC:STATe		
	Turns ON/OFF the measurement between cursors on the trend or queries the current setting.	5-59
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:DC:VALue?		
	Queries the measured value between cursors on the trend.	5-59
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:LINKage		
	Turns ON/OFF the trend cursor link of the waveform parameter measurement or queries the current setting.	5-59
:ANALysis:WPARameter<x>:TRENd:MEASure:MODE		
	Sets the automated measurement mode of the trend of the waveform parameter measurement or queries the current setting.	5-60
:ANALysis:WPARameter<x>:TRENd:VERTical		
	Sets the vertical range of the trend of the waveform parameter measurement or queries the current setting.	5-60
:ANALysis:XY<x>?	Queries all settings related to the XY display function.	5-60
:ANALysis:XY<x>:GATE?	Queries all settings related to the gate function of the XY display.	5-60
:ANALysis:XY<x>:GATE:ALEVel	Sets the active level of the gate of the XY display or queries the current setting.	5-61
:ANALysis:XY<x>:GATE:HYSTeresis<x>	Sets the hysteresis of the gate of the XY display or queries the current setting.	5-61
:ANALysis:XY<x>:GATE:LEVel<x>	Sets the level of the gate of the XY display or queries the current setting.	5-61
:ANALysis:XY<x>:GATE:TRACe	Sets the gate trace of the XY display or queries the current setting.	5-61
:ANALysis:XY<x>:MEASure?	Queries all settings related to the automated measurement of the XY display.	5-61

Command	Function	Page
:ANALysis:XY<x>:MEASure:CURSor?	Queries all settings related to the cursor measurement of the XY display.	5-61
:ANALysis:XY<x>:MEASure:CURSor:XLINkage	Turns ON/OFF the horizontal cursor link on the XY display or queries the current setting.	5-62
:ANALysis:XY<x>:MEASure:CURSor:X<x>?	Queries all settings related to the horizontal cursor of the XY display.	5-62
:ANALysis:XY<x>:MEASure:CURSor:X<x>:POSition	Sets the horizontal cursor position of the XY display or queries the current setting.	5-62
:ANALysis:XY<x>:MEASure:CURSor:X<x>:VALue?	Queries the voltage value at the horizontal cursor of the XY display.	5-62
:ANALysis:XY<x>:MEASure:CURSor:YLINkage	Turns ON/OFF the vertical cursor link on the XY display or queries the current setting.	5-62
:ANALysis:XY<x>:MEASure:CURSor:Y<x>?	Queries all settings related to the vertical cursor of the XY display.	5-62
:ANALysis:XY<x>:MEASure:CURSor:Y<x>:POSition	Sets the vertical cursor position of the XY display or queries the current setting.	5-62
:ANALysis:XY<x>:MEASure:CURSor:Y<x>:VALue?	Queries the voltage value at the vertical cursor of the XY display.	5-63
:ANALysis:XY<x>:MEASure:MODE	Sets the automated measurement mode of the XY display or queries the current setting.	5-63
:ANALysis:XY<x>:MEASure:XYINteg?	Queries all settings related to the integration of the XY display.	5-63
:ANALysis:XY<x>:MEASure:XYINteg:LOOP	Sets the integration mode of the XY display or queries the current setting.	5-63
:ANALysis:XY<x>:MEASure:XYINteg:POLarity	Sets the integration polarity of the XY display or queries the current setting.	5-63
:ANALysis:XY<x>:MEASure:XYINteg:VALue?	Queries the integral value of the XY display.	5-63
:ANALysis:XY<x>:TRANge	Sets the measurement range of the XY display or queries the current setting.	5-63
:ANALysis:XY<x>:WINDow	Sets the measurement source window of the XY display or queries the current setting.	5-63
:ANALysis:XY<x>:XTRace	Sets the X-axis trace of the XY display or queries the current setting.	5-64
:ANALysis:XY<x>:YTRace	Sets the Y-axis trace of the XY display or queries the current setting.	5-64
ASETup Group		
:ASETup:EXECute	Executes auto setup.	5-64
:ASETup:UNDO	Cancels auto setup that has been executed.	5-64
CALibrate Group		
:CALibrate?	Queries all settings related to the calibration.	5-64
:CALibrate:EXECute	Executes calibration.	5-64
:CALibrate:MODE	Turns ON/OFF the auto calibration or queries the current setting.	5-64
CHANnel Group		
:CHANnel<x>?	Queries all settings related to the channel.	5-65
:CHANnel<x>:ASCale[:EXECute]	Executes the auto scaling of the channel.	5-65
:CHANnel<x>:BWIDth	Sets the input filter of the channel or queries the current setting.	5-65
:CHANnel<x>:COUPling	Sets the input coupling of the channel or queries the current setting.	5-65
:CHANnel<x>:DESKew	Sets the skew correction of the channel or queries the current setting.	5-65
:CHANnel<x>:DISPlay	Turns ON/OFF the display of the channel or queries the current setting.	5-65
:CHANnel<x>:INVert	Turns ON/OFF the inverted display of the channel or queries the current setting.	5-65
:CHANnel<x>:LABel?	Queries all settings related to the waveform label of the channel.	5-65
:CHANnel<x>:LABel[:DEFine]	Sets the waveform label of the channel or queries the current setting.	5-65
:CHANnel<x>:LABel:MODE	Turns ON/OFF the waveform label display of the channel or queries the current setting.	5-65
:CHANnel<x>:OCANcel	Turns ON/OFF the offset cancel of the channel or queries the current setting.	5-66
:CHANnel<x>:OFFSet	Sets the offset voltage of the channel or queries the current setting.	5-66
:CHANnel<x>:POSition	Sets the vertical position of the channel or queries the current setting.	5-66

Command	Function	Page
:CHANnel<x>:PROBe?	Queries all settings related to the probe attenuation of the channel.	5-66
:CHANnel<x>:PROBe[:MODE]	Sets the probe attenuation of the channel or queries the current setting.	5-66
:Channal<x>:PROBe:AUTO?	Queries the probe attenuation of the channel when set to AUTO.	5-66
:CHANnel<x>:SELect	Sets the waveform (input/computation) to be assigned to the input channel or queries the current setting.	5-66
:CHANnel<x>:SVALue	Turns ON/OFF the scale display of the channel or queries the current setting.	5-66
:CHANnel<x>:VDIV	Sets the vertical sensitivity (V/div) of the channel or queries the current setting.	5-66
CLEar Group		
:CLEar:ACCumulate	Clears accumulated waveforms.	5-67
:CLEar[:HISTory]	Clears history waveforms.	5-67
:CLEar:SNAP	Clears snapshot waveforms.	5-67
COMMunicate Group		
:COMMunicate?	Queries all settings related to communications.	5-68
:COMMunicate:HEADer	Sets whether to attach a header to the response data or queries the current setting.	5-68
:COMMunicate:LOCKout	Sets or clears local lockout.	5-68
:COMMunicate:OPSE	Sets the overlap command that is to used by the *OPC, *OPC?, and *WAI commands or queries the current setting.	5-68
:COMMunicate:OPSR?	Queries the operation pending status register.	5-68
:COMMunicate:OVERlap	Sets the commands that will operate as overlap commands or queries the current setting.	5-68
:COMMunicate:REMote	Sets remote or local.	5-68
:COMMunicate:STATus?	Queries line-specific status.	5-69
:COMMunicate:VERBose	Sets whether to use abbreviated or unabbreviated form for response data or queries the current setting.	5-69
:COMMunicate:WAIT	Waits for a specified extended event.	5-69
:COMMunicate:WAIT?	Creates the response that is returned when the specified event occurs.	5-69
CURSor Group		
:CURSor?	Queries all settings related to cursor measurements.	5-70
:CURSor:DISPlay	Turns ON/OFF the cursor or queries the current setting.	5-70
:CURSor:HORizontal?	Queries all settings related to the horizontal cursors.	5-70
:CURSor:HORizontal[:BASic]?	Queries all settings related to basic items of the horizontal cursors.	5-70
:CURSor:HORizontal[:BASic]:ALL	Turns ON/OFF all basic items of the horizontal cursors.	5-70
:CURSor:HORizontal[:BASic]:DV?	Queries all settings related to the ΔV measurement of the horizontal cursors.	5-70
:CURSor:HORizontal[:BASic]:DV:STATe	Turns ON/OFF the ΔV measurement of the horizontal cursors or queries the current setting.	5-70
:CURSor:HORizontal[:BASic]:DV:VALue?	Queries the ΔV value of the horizontal cursors.	5-70
:CURSor:HORizontal[:BASic]:LINKage	Turns ON/OFF the horizontal cursor link or queries the current setting.	5-70
:CURSor:HORizontal[:BASic]:V<x>?	Queries all settings related to the horizontal cursor.	5-71
:CURSor:HORizontal[BASic]:V<x>:JUMP	Jumps the horizontal cursor to the center position of the zoom waveform.	5-71
:CURSor:HORizontal[:BASic]:V<x>:POSition	Sets the horizontal cursor position or queries the current setting.	5-71
:CURSor:HORizontal[:BASic]:V<x>:STATe	Turns ON/OFF the horizontal cursor or queries the current setting.	5-71
:CURSor:HORizontal[:BASic]:V<x>:VALue?	Queries the voltage value at the horizontal cursor.	5-71
:CURSor:HORizontal:CALCulation?	Queries all settings related to calculation items of the horizontal cursors.	5-71
:CURSor:HORizontal:CALCulation:ALL	Turns ON/OFF all calculation items of the horizontal cursors.	5-71
:CURSor:HORizontal:CALCulation:DEFine<x>	Sets the equation of the calculation item of the horizontal cursor or queries the current setting.	5-71
:CURSor:HORizontal:CALCulation:STATe<x>	Turns ON/OFF the calculation item of the horizontal cursor or queries the current setting.	5-71

Command	Function	Page
:CURSor:HORizontal:CALCulation:VALue<x>?	Queries the measured value of the calculation item of the horizontal cursor.	5-71
:CURSor:HORizontal:TRACe	Sets the source trace of the horizontal cursor or queries the current setting.	5-72
:CURSor:MARKer?	Queries all settings related to the marker cursors.	5-72
:CURSor:MARKer:CALCulation?	Queries all settings related to calculation items of the marker cursors.	5-72
:CURSor:MARKer:CALCulation:ALL	Turns ON/OFF all calculation items of the marker cursors.	5-72
:CURSor:MARKer:CALCulation:DEFine<x>	Sets the equation of the calculation item of the marker cursors or queries the current setting.	5-72
:CURSor:MARKer:CALCulation:STATe<x>	Turns ON/OFF the calculation item of the marker cursors or queries the current setting.	5-72
:CURSor:MARKer:CALCulation:VALue<x>?	Queries the measured value of the calculation item of the marker cursors.	5-72
:CURSor:MARKer:CM<x>?	Queries all settings related to the marker cursor.	5-72
:CURSor:MARKer:CM<x>:ALL	Turns ON/OFF all items of the marker cursor.	5-73
:CURSor:MARKer:CM<x>:DT<x>?	Queries all settings related to the ΔT measurement of the cursor marker.	5-73
:CURSor:MARKer:CM<x>:DT<x>:STATe	Turns ON/OFF the ΔT measurement of the maker cursor or queries the current setting.	5-73
:CURSor:MARKer:CM<x>:DT<x>:VALue?	Queries the ΔT value of the marker cursor.	5-73
:CURSor:MARKer:CM<x>:DV<x>?	Queries all settings related to the ΔV measurement of the cursor marker.	5-73
:CURSor:MARKer:CM<x>:DV<x>:STATe	Turns ON/OFF the ΔV measurement of the maker cursor or queries the current setting.	5-73
:CURSor:MARKer:CM<x>:DV<x>:VALue?	Queries the ΔV value of the marker cursor.	5-73
:CURSor:MARKer:CM<x>:JUMP	Jumps the marker cursor to the center position of the zoom waveform.	5-73
:CURSor:MARKer:CM<x>:POSition	Sets the marker cursor position or queries the current setting.	5-73
:CURSor:MARKer:CM<x>:T?	Queries all settings related to the time measurement of the marker cursor.	5-73
:CURSor:MARKer:CM<x>:T:STATe	Turns ON/OFF the time measurement of the maker cursor or queries the current setting.	5-74
:CURSor:MARKer:CM<x>:T:VALue?	Queries the time value at the marker cursor position.	5-74
:CURSor:MARKer:CM<x>:TRACe	Sets the source trace of the marker cursor or queries the current setting.	5-74
:CURSor:MARKer:CM<x>:V?	Queries all settings related to the voltage measurement of the marker cursor.	5-74
:CURSor:MARKer:CM<x>:V:STATe	Turns ON/OFF the voltage measurement of the maker cursor or queries the current setting.	5-74
:CURSor:MARKer:CM<x>:V:VALue?	Queries the voltage value at the marker cursor position.	5-74
:CURSor:SERial?	Queries all settings related to the serial cursors.	5-74
:CURSor:SERial:SCURsor<x>?	Queries all settings related to the serial cursor.	5-74
:CURSor:SERial:SCURsor<x>:ACTive	Sets the active level of the serial cursor or queries the current setting.	5-74
:CURSor:SERial:SCURsor<x>:BCOunt	Sets the bit length of the serial cursor or queries the current setting.	5-74
:CURSor:SERial:SCURsor<x>:BITRate	Sets the bit rate of the serial cursor or queries the current setting.	5-75
:CURSor:SERial:SCURsor<x>:BITorder	Sets the bit order of the serial cursor or queries the current setting.	5-75
:CURSor:SERial:SCURsor<x>:FORMat	Sets the display format of the serial cursor or queries the current setting.	5-75
:CURSor:SERial:SCURsor<x>:HYSTeresis	Sets the hysteresis of the serial cursor or queries the current setting.	5-75
:CURSor:SERial:SCURsor<x>:JUMP	Moves the serial cursor to the specified direction.	5-75
:CURSor:SERial:SCURsor<x>:LEVel	Sets the threshold level of the serial cursor or queries the current setting.	5-75
:CURSor:SERial:SCURsor<x>:MODE	Turns ON/OFF the serial cursor or queries the current setting.	5-75
:CURSor:SERial:SCURsor<x>:POSition	Sets the serial cursor position or queries the current setting.	5-75
:CURSor:SERial:SCURsor<x>:TRACE	Sets the trace of the serial cursor or queries the current setting.	5-76
:CURSor:SERial:SCURsor<x>:TRACK	Jumps the serial cursor onto the zoom waveform.	5-76
:CURSor:SERial:SCURsor<x>:VALue?	Queries the measured value of the serial cursor.	5-76
:CURSor:TYPE	Sets the cursor type or queries the current setting.	5-76
:CURSor:VERTical?	Queries all settings related to the vertical cursors.	5-76
:CURSor:VERTical[:BASic]?	Queries all settings related to basic items of the vertical cursors.	5-76
:CURSor:VERTical[:BASic]:ALL	Turns ON/OFF all basic items of the vertical cursors.	5-76
:CURSor:VERTical[:BASic]:DT?	Queries all settings related to the ΔT measurement of the vertical cursors.	5-76
:CURSor:VERTical[:BASic]:DT:STATe	Turns ON/OFF the ΔT measurement of the vertical cursors or queries the current setting.	5-76

Command	Function	Page
:CURSor:VERTical[:BASic]:DT:VALue?	Queries the ΔT value of the vertical cursors.	5-76
:CURSor:VERTical[:BASic]:LINKage	Turns ON/OFF the vertical cursor link or queries the current setting.	5-76
:CURSor:VERTical[:BASic]:PERDt?	Queries all settings related to the 1/ΔT measurement of the vertical cursors.	5-77
:CURSor:VERTical[:BASic]:PERDt:STATe	Turns ON/OFF the 1/ΔT measurement of the vertical cursors or queries the current setting.	5-77
:CURSor:VERTical[:BASic]:PERDt:VALue?	Queries the 1/ΔT value of the vertical cursors.	5-77
:CURSor:VERTical[:BASic]:T<x>?	Queries all settings related to the vertical cursor.	5-77
:CURSor:VERTical[:BASic]:T<x>:JUMP	Jumps the vertical cursor to the center position of the zoom waveform.	5-77
:CURSor:VERTical[:BASic]:T<x>:POSition	Sets the vertical cursor position or queries the current setting.	5-77
:CURSor:VERTical[:BASic]:T<x>:STATe	Turns ON/OFF the vertical cursor or queries the current setting.	5-77
:CURSor:VERTical[:BASic]:T<x>:VALue?	Queries the time value at the vertical cursor.	5-77
:CURSor:VERTical:CALCulation?	Queries all settings related to calculation items of the vertical cursors.	5-77
:CURSor:VERTical:CALCulation:ALL	Turns ON/OFF all calculation items of the vertical cursors.	5-77
:CURSor:VERTical:CALCulation:DEFine<x>	Sets the equation of the calculation item of the vertical cursor or queries the current setting.	5-78
:CURSor:VERTical:CALCulation:STATe<x>	Turns ON/OFF the calculation item of the vertical cursor or queries the current setting.	5-78
:CURSor:VERTical:CALCulation:VALue<x>?	Queries the measured value of the calculation item of the vertical cursor.	5-78
:CURSor:VT?	Queries all settings related to the VT cursor.	5-78
:CURSor:VT[:BASic]?	Queries all settings related to basic items of the VT cursor.	5-78
:CURSor:VT[:BASic]:ALL	Turns ON/OFF all basic items of the VT cursor.	5-78
:CURSor:VT[:BASic]:POSition	Sets the VT cursor position or queries the current setting.	5-78
:CURSor:VT[:BASic]:T?	Queries all settings related to the time value of the VT cursor.	5-78
:CURSor:VT[:BASic]:T:STATe	Turns ON/OFF the time value of the VT cursor or queries the current setting.	5-78
:CURSor:VT[:BASic]:T:VALue?	Queries the time value at the VT cursor.	5-78
:CURSor:VT[:BASic]:V<x>?	Queries all settings related to the voltage value of the VT cursor.	5-79
:CURSor:VT[:BASic]:V<x>:STATe	Turns ON/OFF the voltage value of the VT cursor or queries the current setting.	5-79
:CURSor:VT[:BASic]:V<x>:VALue?	Queries the voltage value at the VT cursor.	5-79
:CURSor:VT:CALCulation?	Queries all settings related to calculation items of the VT cursor.	5-79
:CURSor:VT:CALCulation:ALL	Turns ON/OFF all calculation items of the VT cursor.	5-79
:CURSor:VT:CALCulation:DEFine<x>	Sets the equation of the calculation item of the VT cursor or queries the current setting.	5-79
:CURSor:VT:CALCulation:STATe<x>	Turns ON/OFF the calculation item of the VT cursor or queries the current setting.	5-79
:CURSor:VT:CALCulation:VALue<x>?	Queries the measured value of the calculation item of the VT cursor.	5-79
:CURSor:VT:JUMP	Jumps to the center position of the zoom waveform of the VT cursor.	5-79
DISPlay Group		
:DISPlay?	Queries all settings related to the display.	5-80
:DISPlay:ACCumulate?	Queries all settings related to the accumulated display of waveforms.	5-80
:DISPlay:ACCumulate:GRADe	Sets the accumulate mode or queries the current setting.	5-80
:DISPlay:ACCumulate:MODE	Turns ON/OFF the accumulate mode or queries the current setting.	5-80
:DISPlay:ACCumulate:PERSistence?	Queries all settings related to persistence.	5-80
:DISPlay:ACCumulate:PERSistence:COUNt	Sets the persistence count or queries the current setting.	5-80
:DISPlay:ACCumulate:PERSistence:MODE	Sets the persistence mode or queries the current setting.	5-80
:DISPlay:ACCumulate:PERSistence:TIME	Sets the persistence time or queries the current setting.	5-80

Command	Function	Page
:DISPlay:BLIGht?	Queries all settings related to the backlight.	5-81
:DISPlay:BLIGht:AUTooff	Sets the function that automatically turns the backlight off or queries the current setting.	5-81
:DISPlay:BLIGht:BRIGhtness	Sets the brightness of the backlight or queries the current setting.	5-81
:DISPlay:BLIGht:LCD	Turns ON/OFF the backlight or queries the current setting.	5-81
:DISPlay:BLIGht:TIMeout	Sets the timeout of the backlight or queries the current setting.	5-81
:DISPlay:COLor?	Queries all settings related to the waveform display color.	5-81
:DISPlay:COLor:{CHANnel<x>|MATH<x>|REFerence<x>}	Sets the waveform color or queries the current setting.	5-81
:DISPlay:FORMat	Sets the display format or queries the current setting.	5-81
:DISPlay:GRATicule	Sets the graticule (grid) or queries the current setting.	5-81
:DISPlay:INTENsity?	Queries all settings related to the intensity of the display item.	5-81
:DISPlay:INTENsity[:WAVeform]	Sets the intensity of the waveform or queries the current setting.	5-81
:DISPlay:INTENsity:{CURSor|GRID|MARKer|ZBOX}	Sets the intensity of the display item or queries the current setting.	5-82
:DISPlay:INTERpolate	Sets the display interpolation format or queries the current setting.	5-82
:DISPlay:MAPPing?	Queries all settings related to the waveform mapping to the split screen.	5-82
:DISPlay:MAPPing[:MODE]	Sets the waveform mapping mode for the split screen or queries the current setting.	5-82
:DISPlay:MAPPing:TRACe<x>	Sets the mapping of the waveform to the split screen or queries the current setting.	5-82
FILE Group		
:FILE?	Queries all settings related to the specified storage medium or internal memory.	5-83
:FILE:DIRectory?	Queries all settings related to the specified storage medium.	5-83
:FILE[:DIRectory]:CDIRectory	Sets the current directory or queries the current setting.	5-83
:FILE[:DIRectory]:FREE?	Queries the free disk space in bytes on the specified storage medium.	5-83
:FILE[:DIRectory]:MDIRectory	Creates a directory under the specified directory.	5-84
:FILE:INTernal?	Queries all settings related to the internal memory.	5-84
:FILE:INTernal:RECall:{DMEMory|SETup}[:EXECute]	Executes the loading of specific data from the internal memory.	5-84
:FILE:INTernal:STORe?	Queries all settings related to the saving to the internal memory.	5-84
:FILE:INTernal:STORe:{BINary|DMEMory}?	Queries all settings related to the saving of specific data to the internal memory.	5-84
:FILE:INTernal:STORe:{BINary|DMEMory|SETup}[:EXECute]	Executes the save operation of specific data to the internal memory.	5-84
:FILE:INTernal:STORe:BINary:TRACe	Sets the trace to be saved in binary data to the internal memory or queries the current setting.	5-84
:FILE:INTernal:STORe:DMEMory:TRACe	Sets the trace to be saved in accumulated data to the internal memory or queries the current setting.	5-84
:FILE:INTernal:UNLoad:DMEMory[:EXECute]	Clears the loaded accumulated data.	5-84
:FILE:LOAD?	Queries all settings related to the loading of files on the specified storage medium.	5-84
:FILE:LOAD:{BINary|ZPOLYGON|ZWAVe}?	Queries all settings related to the loading of specific data.	5-84
:FILE:LOAD:{BINary|DMEMory|MASK|SETup|ZPOLygon|ZWAVe}:ABORt	Aborts the loading of the specific data.	5-85
:FILE:LOAD:{BINary|DMEMory|MASK|SETup|ZPOLygon|ZWAVe}[:EXECute]	Executes the loading of the specific data.	5-85
:FILE:LOAD:BINary:REFerence	Sets the load destination of the binary data or queries the current setting.	5-85
:FILE:LOAD:{ZPOLygon|ZWAVe}:ZONE	Sets the load destination of the zone data of specific data or queries the current setting.	5-85
:FILE:SAVE?	Queries all settings related to the saving of the data.	5-85

Command	Function	Page
:FILE:SAVE:{AHIStogram|ASCii|BINary|DMEMory|FFT|FLOat|SBUS|ZWAVe}?		
	Queries all settings related to the saving of the specific data.	5-85
:FILE:SAVE:{AHIStogram|ASCii|BINary|DMEMory|FFT|FLOat|SETup|SBUS|WPARameter|ZWAVe}:ABORt		
	Aborts the save operation of specific data.	5-85
:FILE:SAVE:{AHIStogram|ASCii|BINary|DMEMory|FFT|FLOat|SETup|SBUS|WPARameter|ZWAVe}[:EXECute]		
	Executes the saving of specific data by specifying a file name.	5-86
:FILE:SAVE:{AHIStogram|FFT|SBUS}:ANALysis		
	Sets the trace of specific data to be saved or queries the current setting.	5-86
:FILE:SAVE:ANAMing	Sets the type of auto naming of file names of the saved data or queries the current setting.	5-86
:FILE:SAVE:{ASCii|BINary|FLOat}:COMPression		
	Sets the compression method for saving specific data or queries the current setting.	5-86
:FILE:SAVE:{ASCii|BINary|FLOat}:HISTory		
	Sets whether to save the entire data or the selected data of specific data or queries the current setting.	5-86
:FILE:SAVE:{ASCii|BINary|FLOat}:TRACe		
	Sets the trace of specific data to be saved or queries the current setting.	5-86
:FILE:SAVE:COMMent	Sets the comment of data to be saved or queries the current setting.	5-86
:FILE:SAVE:DMEMory:TRACe	Sets the trace to be saved as accumulated data or queries the current setting.	5-87
:FILE:SAVE:NAME	Sets the name of the data file to be saved or queries the current setting.	5-87
:FILE:SAVE:ZWAVe:ZONE	Sets the zone of the zone data to be saved or queries the current setting.	5-87
GONogo Group		
:GONogo?	Queries all settings related to the GO/NO-GO determination.	5-88
:GONogo:ABORt	Aborts the GO/NO-GO determination.	5-88
:GONogo:ACTion?	Queries all settings related to the action taken when the determination result is NO-GO and the criteria.	5-88
:GONogo:ACTion:BUZZer	Sets whether to sound a buzzer when the determination result is NO-GO or queries the current setting.	5-88
:GONogo:ACTion:HCOPy	Sets whether to print the screen image on the printer when the determination result is NO-GO or queries the current setting.	5-88
:GONogo:ACTion:SAVE	Sets whether to save the waveform data to the storage medium when the determination result is NO-GO or queries the current setting.	5-88
:GONogo:CONDition<x>	Sets the GO/NO-GO determination criteria or queries the current setting.	5-88
:GONogo:COUNt?	Queries the actual number of GO/NO-GO determinations.	5-88
:GONogo:EXECute	Executes the GO/NO-GO determination.	5-88
:GONogo:LOGic	Sets the GO/NO-GO determination logic or queries the current setting.	5-89
:GONogo:MODE	Sets the type of GO/NO-GO determination or queries the current setting.	5-89
:GONogo:NGCount?	Queries the actual number of NO-GOs of the GO/NO-GO determination.	5-89
:GONogo[:SCONdition]?	Queries all settings related to the determination termination condition.	5-89
:GONogo[:SCONdition]:NGCount	Sets the number of NO-GOs that terminates the GO/NO-GO determination or queries the current setting.	5-89
:GONogo[:SCONdition]:STOPcount	Sets the acquisition count that terminates the GO/NO-GO determination or queries the current setting.	5-89
:GONogo:TELecomtest?	Queries all settings related to telecom test determination.	5-89
:GONogo:TELecomtest:SELect<x>?	Queries all settings related to the condition of the telecom test determination.	5-89
:GONogo:TELecomtest:SELect<x>:EYEPattern?		
	Queries all settings related to eye pattern determination of the condition.	5-89
:GONogo:TELecomtest:SELect<x>:EYEPattern:<Parameter>		
	Sets the upper and lower limits of the waveform parameter of the eye pattern determination or queries the current setting.	5-90
:GONogo:TELecomtest:SELect<x>:MASK?	Queries all settings related to the mask determination of the condition.	5-90

Command	Function	Page
:GONogo:TELecomtest:SELect<x>:MASK:ELEMent<x>?	Queries all settings related to the element used in the mask determination.	5-90
:GONogo:TELecomtest:SELect<x>:MASK:ELEMent<x>:PSPCount	Sets the upper and lower limits of the error rate for the number of sampled data points of the element or queries the current setting.	5-90
:GONogo:TELecomtest:SELect<x>:MASK:ELEMent<x>:PWCount	Sets the upper and lower limits of the error rate for the number of acquisitions of the element or queries the current setting.	5-90
:GONogo:TELecomtest:SELect<x>:MASK:ELEMent<x>:SPCount	Sets the upper and lower limits of the number of sampled data points for the element that results in error or queries the current setting.	5-91
:GONogo:TELecomtest:SELect<x>:MASK:ELEMent<x>:WCOunt	Sets the upper and lower limits of the number of acquisitions for the element that results in error or queries the current setting.	5-91
:GONogo:ZPARameter?	Queries all settings related to zone/parameter determination.	5-91
:GONogo:ZPARameter:SELect<x>?	Queries all settings related to the condition of the zone/parameter determination.	5-91
:GONogo:ZPARameter:SELect<x>:MODE	Sets the mode of the condition or queries the current setting.	5-91
:GONogo:ZPARameter:SELect<x>:PARameter?	Queries all settings related to the condition parameter.	5-92
:GONogo:ZPARameter:SELect<x>:PARameter:CATegory	Sets the parameter category or queries the current setting.	5-92
:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>?	Queries all settings related to the FFT determination.	5-92
:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:CALCulation<x>	Sets the upper and lower limits of the calculation item of the FFT determination or queries the current setting.	5-92
:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:PEAK?	Queries all settings related to the peak value of the FFT determination.	5-92
:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:PEAK:DFREquency	Sets the upper and lower limits between the peak frequencies of the FFT determination or queries the current setting.	5-92
:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:PEAK:DV	Sets the upper and lower limits between the peak voltages of the FFT determination or queries the current setting.	5-93
:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:PEAK:FREQuency<x>	Sets the upper and lower limits of the peak frequency of the FFT determination or queries the current setting.	5-93
:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:PEAK:V<x>	Sets the upper and lower limits of the peak voltage of the FFT determination or queries the current setting.	5-93
:GONogo:ZPARameter:SELect<x>:PARameter:MEASure?	Queries all settings related to the determination using automated measurement of waveform parameters.	5-93
:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:CALCulation<x>	Sets the upper and lower limits of the calculation item of the measure determination or queries the current setting.	5-93
:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:STATistics	Sets the statistical value of the measure determination or queries the current setting.	5-94
:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:TRACe<x>?	Queries all settings related to the trace of the measure determination.	5-94
:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>?	Queries all settings related to the area of the measure determination.	5-94
:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>:TYPE?	Queries all settings related to the waveform parameters of the measure determination.	5-94

Command	Function	Page
:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>:TYPE:<Parameter>	Sets the upper and lower limits of the waveform parameter of the measure determination or queries the current setting.	5-94
:GONogo:ZPARameter:SELect<x>:PARameter:XY<x>?	Queries all settings related to the XY determination.	5-95
:GONogo:ZPARameter:SELect<x>:PARameter:XY<x>:XYINteg	Sets the upper and lower limits integral value of the XY determination or queries the current setting.	5-95
:GONogo:ZPARameter:SELect<x>:RECTangle?	Queries all settings related to the rectangle determination.	5-95
:GONogo:ZPARameter:SELect<x>:RECTangle:HORizontal	Sets the horizontal position of the rectangle used in the rectangle determination or queries the current setting.	5-95
:GONogo:ZPARameter:SELect<x>:RECTangle:VERTical	Sets the vertical position of the rectangle used in the rectangle determination or queries the current setting.	5-95
:GONogo:ZPARameter:SELect<x>:TRACe	Sets the source trace of the zone/parameter determination or queries the current setting.	5-95
:GONogo:ZPARameter:SELect<x>:WAVE?	Queries all settings related to the wave determination.	5-95
:GONogo:ZPARameter:SELect<x>:WAVE:EDIT<x>:EXIT	Exits the zone edit menu of the wave determination.	5-96
:GONogo:ZPARameter:SELect<x>:WAVE:EDIT<x>:PART	Sets the editing of the portion of the zone of the wave determination.	5-96
:GONogo:ZPARameter:SELect<x>:WAVE:EDIT<x>:WHOLe	Sets the editing of the entire zone of the waveform zone.	5-96
:GONogo:ZPARameter:SELect<x>:WAVE:TRANge	Sets the determination range of the zone determination or queries the current setting.	5-96
:GONogo:ZPARameter:SELect<x>:WINDow	Sets the source window of the zone determination or queries the current setting.	5-96
HCOPy Group		
:HCOPy?	Queries all settings related to the output of screen data.	5-97
:HCOPy:ABORt	Aborts data output and paper feeding.	5-97
:HCOPy:DIRection	Sets the data output destination or queries the current setting.	5-97
:HCOPy:EXECute	Executes the data output.	5-97
:HCOPy:EXTPrinter?	Queries all settings related to the external printer output.	5-97
:HCOPy:EXTPrinter:TONE	Sets the half tone of the external printer output or queries the current setting.	5-97
:HCOPy:EXTPrinter:TYPE	Sets the type of output commands to send to the external printer or queries the current setting.	5-97
:HCOPy:FILE?	Queries all settings related to file output.	5-97
:HCOPy:FILE:FORMat	Sets the file output image format or queries the current setting.	5-97
:HCOPy:FILE:SAVE?	Queries all settings related to the saving of file output.	5-97
:HCOPy:FILE:SAVE:ANAMing	Sets the type of auto naming of save destination file names or queries the current setting.	5-98
:HCOPy:FILE:SAVE:CDIRectory	Sets the save destination directory name or queries the current setting.	5-98
:HCOPy:FILE:SAVE:NAME	Sets the save destination file name or queries the current setting.	5-98
:HCOPy:PRINter?	Queries all settings related to the built-in printer output.	5-98
:HCOPy:PRINter:HRMode	Turns ON/OFF the harmonic analysis mode of the built-in printer output or queries the current setting.	5-98
HISTory Group		
:HISTory?	Queries all settings related to the history function.	5-99
:HISTory[:CURRent]?	Queries all settings related to the history function of the current waveform (CH1 to 4, M1 to 8).	5-99
:HISTory[:CURRent]:DISPlay	Sets the start number and end number of the display record of the history waveform or queries the current setting.	5-99

Command	Function	Page
:HISTory[:CURRent]:DMODe	Sets the display mode of the history waveform or queries the current setting.	5-100
:HISTory[:CURRent]:MODE	Sets the highlight display mode of the history waveform or queries the current setting.	5-100
:HISTory[:CURRent]:RECord	Sets the target record of the history waveform or queries the current setting.	5-100
:HISTory[:CURRent]:RECord? MINimum	Queries the minimum record number of the history waveform.	5-100
:HISTory[:CURRent]:REPLay?	Queries all settings related to the replay function of the history function.	5-100
:HISTory[:CURRent]:REPLay:JUMP	Jumps the history waveform to the specified record number.	5-100
:HISTory[:CURRent]:REPLay:SPEed	Sets the replay speed of the history waveform or queries the current setting.	5-100
:HISTory[:CURRent]:REPLay:STARt	Starts the replay of the history waveform.	5-100
:HISTory[:CURRent]:REPLay:STOP	Stops the replay of the history waveform.	5-100
:HISTory[:CURRent][:SEARch]?	Queries all settings related to the history search function.	5-101
:HISTory[:CURRent][:SEARch]:ABORt	Aborts the history search.	5-101
:HISTory[:CURRent][:SEARch]:EXECute	Executes the history search.	5-101
:HISTory[:CURRent][:SEARch]:LOGic	Sets the history search logic or queries the current setting.	5-101
:HISTory[:CURRent][:SEARch]:RESet	Resets the search conditions of the history search.	5-101
:HISTory[:CURRent][:SEARch]:SELect<x>?	Queries all settings related to the history search condition.	5-101
:HISTory[:CURRent][:SEARch]:SELect<x>:CONDition	Sets determination criteria of the history search condition or queries the current setting.	5-101
:HISTory[:CURRent][:SEARch]:SELect<x>:MODE	Sets the mode of the history search condition or queries the current setting.	5-101
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter?	Queries all settings related to the parameter of the history search condition.	5-102
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:CATegory	Sets the parameter category or queries the current setting.	5-102
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:FFT<x>?	Queries all settings related to the FFT search.	5-102
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:FFT<x>:CALCulation<x>	Sets the upper and lower limits of the calculation item of the FFT search or queries the current setting.	5-102
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:FFT<x>:PEAK?	Queries all settings related to the peak value of the FFT search.	5-102
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:FFT<x>:PEAK:DFREquency	Sets the upper and lower limits between the peak frequencies of the FFT search or queries the current setting.	5-103
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:FFT<x>:PEAK:DV	Sets the upper and lower limits between the peak voltages of the FFT search or queries the current setting.	5-103
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:FFT<x>:PEAK:FREquency<x>	Sets the upper and lower limits of the peak frequency of the FFT search or queries the current setting.	5-103
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:FFT<x>:PEAK:V<x>	Sets the upper and lower limits of the peak voltage of the FFT search or queries the current setting.	5-103
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:MEASure?	Queries all settings related to the measure search.	5-103
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:MEASure:CALCulation<x>	Sets the upper and lower limits of the calculation item of the measure search or queries the current setting.	5-104
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:MEASure:TRACe<x>?	Queries all settings related to the trace of the measure search.	5-104
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>?	Queries all settings related to the area of the measure search.	5-104
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>:TYPE?	Queries all settings related to the waveform parameters of the measure search.	5-104

Command	Function	Page
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>:TYPE:<Parameter>	Sets the upper and lower limits of the waveform parameter of the measure search or queries the current setting.	5-105
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:XY<x>?	Queries all settings related to the XY search.	5-105
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:XY<x>:XYINteg	Sets the upper and lower limits integral value of the XY search or queries the current setting.	5-105
:HISTory[:CURRent][:SEARch]:SELect<x>:RECTangle?	Queries all settings related to the rectangle search.	5-105
:HISTory[:CURRent][:SEARch]:SELect<x>:RECTangle:HORizontal	Sets the horizontal position of the rectangle used in the rectangle search or queries the current setting.	5-105
:HISTory[:CURRent][:SEARch]:SELect<x>:RECTangle:VERTical	Sets the vertical position of the rectangle used in the rectangle search or queries the current setting.	5-106
:HISTory[:CURRent][:SEARch]:SELect<x>:TRACe	Sets the source trace of the history search or queries the current setting.	5-106
:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE?	Queries all settings related to the wave search.	5-106
:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE:EDIT<x>:EXIT	Exits the zone edit menu of the wave search.	5-106
:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE:EDIT<x>:PART	Sets the editing of the portion of the zone of the wave search.	5-106
:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE:EDIT<x>:WHOLe	Sets the editing of the entire zone of the wave search.	5-106
:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE:TRANge	Sets the range over which to perform the wave search or queries the current setting.	5-107
:HISTory[:CURRent][:SEARch]:SELect<x>:WINDow	Sets the source window of the history search or queries the current setting.	5-107
:HISTory[:CURRent]:TIME?	Queries the time of the source record number of the history waveform.	5-107
:HISTory:REFerence<x>?	Queries all settings related to the history function of the reference.	5-107
:HISTory:REFerence<x>:DMODe	Sets the display mode of the history waveform of the reference or queries the current setting.	5-107
:HISTory:REFerence<x>:MODE	Sets the highlight display mode of the history waveform of the reference or queries the current setting.	5-107
:HISTory:REFerence<x>:RECord	Sets the source record of the history waveform of the reference or queries the current setting.	5-107
:HISTory:REFerence<x>:RECord? MINimum	Queries the minimum record number of the history waveform of the reference.	5-107
:HISTory:REFerence<x>:REPLay?	Queries all settings related to the replay function of the history function of the reference.	5-107
:HISTory:REFerence<x>:REPLay:JUMP	Jumps to the specified record number of the history waveform of the reference.	5-108
:HISTory:REFerence<x>:REPLay:SPEed	Sets the replay speed of the history waveform of the reference or queries the current setting.	5-108
:HISTory:REFerence<x>:REPLay:STARt	Starts the replay of the history waveform of the reference.	5-108
:HISTory:REFerence<x>:REPLay:STOP	Stops the replay of the history waveform of the reference.	5-108
:HISTory:REFerence<x>:TIME?	Queries the time of the source record number of the reference waveform.	5-108
IMAGe Group		
:IMAGe?	Queries all settings related to the output of screen image data.	5-109
:IMAGe:FORMat	Sets the output format of the screen image data or queries the current setting.	5-109
:IMAGe:SEND?	Queries the screen image data.	5-109
:IMAGe:TONE	Sets the tone of the screen image data or queries the current setting.	5-109

Command	Function	Page
INITialize Group		
:INITialize:EXECute	Executes the initialization.	5-109
:INITialize:UNDO	Cancels the initialization that has been executed.	5-109
MATH Group		
:MATH<x>?	Queries all settings related to the computation.	5-110
:MATH<x>:DISPlay	Turns ON/OFF the computed waveform or queries the current setting.	5-110
:MATH<x>:ECOunt?	Queries all settings related to the edge count computation.	5-110
:MATH<x>:ECOunt:HYSTeresis	Sets the hysteresis of the edge detection level of the edge count computation or queries the current setting.	5-110
:MATH<x>:ECOunt:POLarity	Sets the edge detection polarity of the edge count computation or queries the current setting.	5-110
:MATH<x>:FILTer?	Queries all settings related to filters.	5-110
:MATH<x>:FILTer:DELay?	Queries all settings related to the delay computation.	5-110
:MATH<x>:FILTer:DELay:TIME	Sets the delay value of the delay computation or queries the current setting.	5-110
:MATH<x>:FILTer:IIR?	Queries all settings related to the IIR filter computation.	5-111
:MATH<x>:FILTer:IIR:FORDer	Sets the filter order of the IIR filter computation or queries the current setting.	5-111
:MATH<x>:FILTer:IIR:HIPass?	Queries all settings related to the IIR high pass filter computation.	5-111
:MATH<x>:FILTer:IIR:HIPass:COFF	Sets the cutoff frequency of the IIR high pass filter computation or queries the current setting.	5-111
:MATH<x>:FILTer:IIR:LOWPass?	Queries all settings related to the IIR low pass filter computation.	5-111
:MATH<x>:FILTer:IIR:LOWPass:COFF	Sets the cutoff frequency of the IIR low pass filter computation or queries the current setting.	5-111
:MATH<x>:FILTer:MAVG?	Queries all settings related to the moving average computation.	5-111
:MATH<x>:FILTer:MAVG:WEIGht	Sets the weight of the moving average computation or queries the current setting.	5-111
:MATH<x>:FILTer:RESCaling?	Queries all settings related to the rescaling of the filter.	5-111
:MATH<x>:FILTer:RESCaling:AVALue	Sets rescaling coefficient A of the filter or queries the current setting.	5-111
:MATH<x>:FILTer:RESCaling:BVALue	Sets rescaling offset B of the filter or queries the current setting.	5-112
:MATH<x>:FILTer:TYPE	Sets the filter type or queries the current setting.	5-112
:MATH<x>:INTegral?	Queries all settings related to the integral computation.	5-112
:MATH<x>:INTegral:PSCaling?	Queries all settings related to the pre-scaling of the integral computation.	5-112
:MATH<x>:INTegral:PSCaling:AVALue	Sets pre-scaling coefficient A of the integral computation or queries the current setting.	5-112
:MATH<x>:INTegral:PSCaling:BVALue	Sets pre-scaling offset B of the integral computation or queries the current setting.	5-112
:MATH<x>:INTegral:RESCaling?	Queries all settings related to the rescaling of the integral computation.	5-112
:MATH<x>:INTegral:RESCaling:AVALue	Sets rescaling coefficient A of the integral computation or queries the current setting.	5-112
:MATH<x>:INTegral:RESCaling:BVALue	Sets rescaling offset B of the integral computation or queries the current setting.	5-112
:MATH<x>:INVert	Turns ON/OFF the inverted display of the computed waveform or queries the current setting.	5-112
:MATH<x>:IPOint?	Queries all settings related to the computation reference point.	5-113
:MATH<x>:IPOint:JUMP	Moves the computation reference point to the specified position.	5-113
:MATH<x>:IPOint:POSition	Sets the computation reference point or queries the current setting.	5-113
:MATH<x>:LABel?	Queries all settings related to the label of the computed waveform.	5-113
:MATH<x>:LABel[:DEFine]	Sets the label of the computed waveform or queries the current setting.	5-113
:MATH<x>:LABel:MODE	Turns ON/OFF the label display of the computed waveform or queries the current setting.	5-113
:MATH<x>:MINus?	Queries all settings related to the subtraction.	5-113
:MATH<x>:MINus:PSCaling<x>?	Queries all settings related to the pre-scaling of the subtraction.	5-113
:MATH<x>:MINus:PSCaling<x>:AVALue	Sets pre-scaling coefficient A of the subtraction or queries the current setting.	5-113
:MATH<x>:MINus:PSCaling<x>:BVALue	Sets pre-scaling offset B of the subtraction or queries the current setting.	5-113
:MATH<x>:MINus:RESCaling?	Queries all settings related to the rescaling of the subtraction.	5-114
:MATH<x>:MINus:RESCaling:AVALue	Sets rescaling coefficient A of the subtraction or queries the current setting.	5-114
:MATH<x>:MINus:RESCaling:BVALue	Sets rescaling offset B of the subtraction or queries the current setting.	5-114

Command	Function	Page
:MATH<x>:MULTiple?	Queries all settings related to the multiplication.	5-114
:MATH<x>:MULTiple:PSCaling<x>?	Queries all settings related to the pre-scaling of the multiplication.	5-114
:MATH<x>:MULTiple:PSCaling<x>:AVALue	Sets pre-scaling coefficient A of the multiplication or queries the current setting.	5-114
:MATH<x>:MULTiple:PSCaling<x>:BVALue	Sets pre-scaling offset B of the multiplication or queries the current setting.	5-114
:MATH<x>:MULTiple:RESCaling?	Queries all settings related to the rescaling of the multiplication.	5-114
:MATH<x>:MULTiple:RESCaling:AVALue	Sets rescaling coefficient A of the multiplication or queries the current setting.	5-114
:MATH<x>:MULTiple:RESCaling:BVALue	Sets rescaling offset B of the multiplication or queries the current setting.	5-115
:MATH<x>:OPERation	Sets the operator or queries the current setting.	5-115
:MATH<x>:PLUS?	Queries all settings related to the addition.	5-115
:MATH<x>:PLUS:PSCaling<x>?	Queries all settings related to the pre-scaling of the addition.	5-115
:MATH<x>:PLUS:PSCaling<x>:AVALue	Sets pre-scaling coefficient A of the addition or queries the current setting.	5-115
:MATH<x>:PLUS:PSCaling<x>:BVALue	Sets pre-scaling offset B of the addition or queries the current setting.	5-115
:MATH<x>:PLUS:RESCaling?	Queries all settings related to the rescaling of the addition.	5-115
:MATH<x>:PLUS:RESCaling:AVALue	Sets rescaling coefficient A of the addition or queries the current setting.	5-115
:MATH<x>:PLUS:RESCaling:BVALue	Sets rescaling offset B of the addition or queries the current setting.	5-115
:MATH<x>:POSition	Sets the vertical position of the computed waveform or queries the current setting.	5-116
:MATH<x>:SCALe?	Queries all settings related to scaling.	5-116
:MATH<x>:SCALe:CENTer	Sets the offset of the computed waveform or queries the current setting.	5-116
:MATH<x>:SCALe:MODE	Sets the scaling mode or queries the current setting.	5-116
:MATH<x>:SCALe:SENSitivity	Sets the vertical sensitivity of the computed waveform or queries the current setting.	5-116
:MATH<x>:SELect	Sets the display option or queries the current setting.	5-116
:MATH<x>:SVALue	Turns ON/OFF the scale value display or queries the current setting.	5-116
:MATH<x>:THReshold<x>	Sets the edge detection level of the count computation or queries the current setting.	5-116
:MATH<x>:UNIT?	Queries all settings related to the computation unit.	5-116
:MATH<x>:UNIT[:DEFine]	Sets the computation unit or queries the current setting.	5-116
:MATH<x>:UNIT:MODE	Sets the automatic/manual addition of the computation unit or queries the current setting.	5-117
MEASure Group		
:MEASure?	Queries all settings related to the automated measurement of waveform parameters.	5-118
:MEASure:CALCulation?	Queries all settings related to calculation items.	5-118
:MEASure:CALCulation:ALL	Turns ON/OFF all calculation items.	5-118
:MEASure:CALCulation:COUNt<x>?	Queries the statistical processing count of the calculation item.	5-118
:MEASure:CALCulation:DEFine<x>	Sets the equation of the calculation item or queries the current setting.	5-118
:MEASure:CALCulation:{MAXimum<x>|MEAN<x>|MINimum<x>|SDEViation<x>}?	Queries the statistical value of the calculation item.	5-118
:MEASure:CALCulation:STATe<x>	Turns ON/OFF the calculation item or queries the current setting.	5-118
:MEASure:CALCulation:VALue<x>?	Queries the automated measured value of the calculation item.	5-119
:MEASure:CONTinuous?	Queries all settings related to the continuous statistical processing.	5-119
:MEASure:CONTinuous:COUNt	Sets the continuous statistical processing count or queries the current setting.	5-119
:MEASure:CONTinuous:RESTart	Restarts the continuous statistical processing.	5-119
:MEASure:CYCLe?	Queries all settings related to the cycle statistical processing.	5-119
:MEASure:CYCLe:ABORt	Aborts the execution of the cycle statistical processing.	5-119
:MEASure:CYCLe:EXECute	Executes the cycle statistical processing.	5-119
:MEASure:CYCLe:TRACe	Sets the cycle source trace of the continuous statistical processing count or queries the current setting.	5-119
:MEASure:DISPlay	Turns ON/OFF the display of the automated measurement of waveform parameters or queries the current setting.	5-119
:MEASure:HISTory:ABORt	Aborts the execution of the statistical processing of the history data.	5-119
:MEASure:HISTory:EXECute	Executes the statistical processing of the history data.	5-119

Command	Function	Page
:MEASure:MODE	Sets the mode of the automated measurement of waveform parameters or queries the current setting.	5-119
:MEASure:THReshold?	Queries all settings related to the threshold levels of the automated measurement of waveform parameters.	5-120
:MEASure:THReshold:TRACe<x>?	Queries the threshold levels of the trace.	5-120
:MEASure:THReshold:TRACe<x>:AUTO	Sets the detection mode when the auto setting of the threshold level is enabled or queries the current setting.	5-120
:MEASure:THReshold:TRACe<x>:LHYSteresis?	Queries all settings related to the level and hysteresis of the threshold level.	5-120
:MEASure:THReshold:TRACe<x>:LHYSteresis:HYSTeresis	Sets the hysteresis of the threshold level or queries the current setting.	5-120
:MEASure:THReshold:TRACe<x>:LHYSteresis:LEVel	Sets the level of the threshold level or queries the current setting.	5-121
:MEASure:THReshold:TRACe<x>:MODE	Sets the setup mode of the threshold level or queries the current setting.	5-121
:MEASure:THReshold:TRACe<x>:ULOWer?	Queries all settings related to the upper and lower limits of the threshold level.	5-121
:MEASure:THReshold:TRACe<x>:ULOWer:RANGe	Sets the upper and lower limits of the threshold level or queries the current setting.	5-121
:MEASure:TRACe<x>?	Queries all settings related to the trace.	5-121
:MEASure:TRACe<x>:AREA<x>?	Queries all settings related to the area.	5-122
:MEASure:TRACe<x>:AREA<x>:ALL	Turns ON/OFF all waveform parameters.	5-122
:MEASure:TRACe<x>:AREA<x>:<Parameter>?	Queries all settings related to the waveform parameter.	5-122
:MEASure:TRACe<x>:AREA<x>:<Parameter>:COUNt?	Queries the continuous statistical processing count of the waveform parameter.	5-122
:MEASure:TRACe<x>:AREA<x>:<Parameter>:{MAXimum|MEAN|MINimum|SDEViation}?	Queries the statistical value of the waveform parameter.	5-123
:MEASure:TRACe<x>:AREA<x>:<Parameter>:STATe	Turns ON/OFF the waveform parameter or queries the current setting.	5-123
:MEASure:TRACe<x>:AREA<x>:<Parameter>:VALue?	Queries the automated measured value of the waveform parameter.	5-123
:MEASure:TRACe<x>:AREA<x>:DELay:MEASure?	Queries all settings related to the measurement conditions of the source waveform of the delay measurement between channels.	5-123
:MEASure:TRACe<x>:AREA<x>:DELay:MEASure:COUNt	Sets the edge detection count of the source waveform of the delay measurement between channels or queries the current setting.	5-124
:MEASure:TRACe<x>:AREA<x>:DELay:MEASure:POLarity	Sets the polarity of the source waveform of the delay measurement between channels or queries the current setting.	5-124
:MEASure:TRACe<x>:AREA<x>:DELay:REFerence?	Queries all settings related to the reference waveform of the delay measurement between channels.	5-124
:MEASure:TRACe<x>:AREA<x>:DELay:REFerence:COUNt	Sets the edge detection count of the reference waveform of the delay measurement between channels or queries the current setting.	5-124
:MEASure:TRACe<x>:AREA<x>:DELay:REFerence:POLarity	Sets the polarity of the reference waveform of the delay measurement between channels or queries the current setting.	5-124
:MEASure:TRACe<x>:AREA<x>:DELay:REFerence:TRACe	Sets the trace of the reference waveform of the delay measurement between channels or queries the current setting.	5-124
:MEASure:TRACe<x>:AREA<x>:DELay:SOURce	Sets the reference of the delay measurement between channels or queries the current setting.	5-125

Command	Function	Page
:MEASure:TRACe<x>:AREA<x>:DPRoximal?		
	Queries all settings related to the distal and proximal values.	5-125
:MEASure:TRACe<x>:AREA<x>:DPRoximal:MODE		
	Sets the unit of the distal and proximal values or queries the current setting.	5-125
:MEASure:TRACe<x>:AREA<x>:DPRoximal:PERCent		
	Sets the distal and proximal values as a percentage or queries the current setting.	5-125
:MEASure:TRACe<x>:AREA<x>:DPRoximal:UNIT		
	Sets the distal and proximal values in the specified unit or queries the current setting.	5-125
:MEASure:TRANge<x>	Sets the measurement range or queries the current setting.	5-125
:MEASure:WAIT?	Waits for the completion of the automated measurement with a timeout option.	5-126
:MEASure:WINDow<x>	Sets the measurement source window of the area or queries the current setting.	5-126
REFerence Group		
:REFerence<x>?	Queries all settings related to the reference.	5-127
:REFerence<x>:DISPlay	Turns ON/OFF the display of the reference or queries the current setting.	5-127
:REFerence<x>:INVert	Sets the inverted display of the reference or queries the current setting.	5-127
:REFerence<x>:LABel?	Queries all settings related to the waveform label of the reference.	5-127
:REFerence<x>:LABel[:DEFine]	Sets the waveform label of the reference or queries the current setting.	5-127
:REFerence<x>:LABel:MODE	Turns ON/OFF the waveform label display of the reference or queries the current setting.	5-127
:REFerence<x>:LOAD	Loads the waveform to the reference.	5-127
:REFerence<x>:POSition	Sets the vertical position of the reference or queries the current setting.	5-127
:REFerence<x>:SELect	Sets the waveform (computation or reference) to the computation channel or queries the current setting.	5-127
:REFerence<x>:SVALue	Turns ON/OFF the scale display of the reference or queries the current setting.	5-127
SEARch Group		
:SEARch<x>?	Queries all settings related to the search function.	5-128
:SEARch<x>:ABORt	Aborts the search.	5-128
:SEARch<x>:CLOCk?	Queries all settings related to the clock channel.	5-128
:SEARch<x>:CLOCk:POLarity	Sets the polarity of the clock channel or queries the current setting.	5-128
:SEARch<x>:CLOCk:SOURce	Sets the clock trace of the search or queries the current setting.	5-128
:SEARch<x>:DECimation	Sets the decimation detection of the skip mode or queries the current setting.	5-128
:SEARch<x>:EXECute	Executes the search.	5-129
:SEARch<x>:HOLDoff	Sets the hold off detection or queries the current setting.	5-129
:SEARch<x>:LOGic	Sets the search logic or queries the current setting.	5-129
:SEARch<x>:POLarity	Sets the search polarity or queries the current setting.	5-129
:SEARch<x>:SELect	Sets the detection waveform number of the search function or queries the current setting.	5-129
:SEARch<x>:SELect? MAXimum	Queries the detection count of the search function.	5-129
:SEARch<x>:SMODe	Sets the skip mode or queries the current setting.	5-129
:SEARch<x>:SPATtern?	Queries all settings related to the serial pattern search.	5-129
:SEARch<x>:SPATtern:CLOCk?	Queries all settings related to clock of the serial pattern search.	5-129
:SEARch<x>:SPATtern:CLOCk:MODE	Enables/Disables the clock of the serial pattern search or queries the current setting.	5-130
:SEARch<x>:SPATtern:CLOCk:POLarity	Sets the polarity of the clock trace of the serial pattern search or queries the current setting.	5-130
:SEARch<x>:SPATtern:CLOCk:SOURce	Sets the clock trace of the serial pattern search or queries the current setting.	5-130
:SEARch<x>:SPATtern:CS	Enables/Disables the chip select of the serial pattern search or queries the current setting.	5-130
:SEARch<x>:SPATtern:LATCh?	Queries all settings related to latch of the serial pattern search.	5-130
:SEARch<x>:SPATtern:LATCh:POLarity	Sets the polarity of the latch trace of the serial pattern search or queries the current setting.	5-130

Command	Function	Page
:SEARch<x>:SPATtern:LATCh:TRACe	Sets the latch trace of the serial pattern search or queries the current setting.	5-130
:SEARch<x>:SPATtern[:SETup]?	Queries all settings related to setup of the serial pattern search.	5-130
:SEARch<x>:SPATtern[:SETup]:BITRate	Sets the bit rate of the serial pattern search or queries the current setting.	5-131
:SEARch<x>:SPATtern[:SETup]:CLEar	Clears the entire pattern of the serial pattern search.	5-131
:SEARch<x>:SPATtern[:SETup]:DATA?	Queries all settings related to data of the serial pattern search.	5-131
:SEARch<x>:SPATtern[:SETup]:DATA:ACTive	Sets the active level of the data trace of the serial pattern search or queries the current setting.	5-131
:SEARch<x>:SPATtern[:SETup]:DATA:TRACe	Sets the data trace of the serial pattern search or queries the current setting	5-131
:SEARch<x>:SPATtern[:SETup]:HEXA	Sets the pattern of the serial pattern search in hexadecimal notation.	5-131
:SEARch<x>:SPATtern[:SETup]:PATTern	Sets the pattern of the serial pattern search in binary notation or queries the current setting.	5-131
:SEARch<x>:SPOint	Sets the search start position or queries the current setting.	5-131
:SEARch<x>:STRace	Sets the search source trace or queries the current setting.	5-131
:SEARch<x>:TRACe<x>?	Queries all settings related to the search conditions of the trace.	5-132
:SEARch<x>:TRACe<x>:CONDition	Sets the condition to be satisfied for the trace or queries the current setting.	5-132
:SEARch<x>:TRACe<x>:HYSTeresis	Sets the hysteresis of the trace or queries the current setting.	5-132
:SEARch<x>:TRACe<x>:LEVel	Sets the threshold level of the trace or queries the current setting.	5-132
:SEARch<x>:TYPE	Sets the search type or queries the current setting.	5-132
:SEARch<x>:WIDTh?	Queries all settings related to the pulse width search.	5-132
:SEARch<x>:WIDTh:MODE	Sets the pulse width determination mode or queries the current setting.	5-132
:SEARch<x>:WIDTh:TIME<x>	Sets the pulse width of the pulse width search or queries the current setting.	5-132
:SEARch<x>:WIDTh:TYPE	Sets the pulse width search type or queries the current setting.	5-132
SNAP Group		
:SNAP	Executes the snapshot.	5-133
SSTart Group		
:SSTart?	Executes the single start of the trigger mode.	5-133
STARt Group		
:STARt	Starts waveform acquisition.	5-133
STATus Group		
:STATus?	Queries all settings related to the communication status function.	5-134
:STATus:CONDition?	Queries the contents of the condition register.	5-134
:STATus:EESE	Sets the extended event enable register or queries the current setting.	5-134
:STATus:EESR?	Queries the content of the extended event register and clears the register.	5-134
:STATus:ERRor?	Queries the code and message of the error that occurred.	5-134
:STATus:FILTer<x>	Sets the transition filter or queries the current setting.	5-134
:STATus:QENable	Sets whether to store messages other than errors to the error queue or queries the current setting.	5-134
:STATus:QMESsage	Sets whether to attach a message description to the response to the :STATus:ERRor? query or queries the current setting.	5-134
:STATus:SPOLl?	Executes serial polling.	5-134
STOP Group		
:STOP	Stops waveform acquisition.	5-135
SYSTem Group		
:SYSTem?	Queries all settings related to the system.	5-136
:SYSTem:BATTery?	Queries the condition of the internal lithium battery.	5-136
:SYSTem:CLICk	Turns ON/OFF the click sound or queries the current setting.	5-136
:SYSTem:CLOCk?	Queries all settings related to the date, time, and time difference with respect to GMT.	5-136

Command	Function	Page
:SYSTem:CLOCk:DTIMe	Sets the date, time, and time difference with respect to GMT or queries the current setting.	5-136
:SYSTem:CLOCk:MODE	Turns ON/OFF the date, time, and time difference with respect to GMT or queries the current setting.	5-136
:SYSTem:FORMat:IMEMory[:EXECute]	Formats the internal memory.	5-136
:SYSTem:FORMat:IHDD[:EXECute]	Formats the internal hard disk.	5-136
:SYSTem:FORMat:SDELete[:EXECute]	Clears and formats the internal memory.	5-136
:SYSTem:LANGuage	Sets the message language or queries the current setting.	5-136
:SYSTem:MLANguage	Sets the menu language or queries the current setting.	5-136
:SYSTem:OVERview	Displays system information.	5-136
TELecomtest Group		
:TELecomtest?	Queries all settings related to the telecom test.	5-137
:TELecomtest:CATegory	Sets the telecom test type or queries the current setting.	5-137
:TELecomtest:DISPlay	Turns ON/OFF the telecom test display or queries the current setting.	5-137
:TELecomtest:EYEPattern?	Queries all settings related to the eye pattern.	5-137
:TELecomtest:EYEPattern:ALL	Turns ON/OFF all eye pattern parameters.	5-137
:TELecomtest:EYEPattern:<Parameter>?	Queries all settings related to the waveform parameter of the eye pattern.	5-137
:TELecomtest:EYEPattern:<Parameter>:STATe	Turns ON/OFF the waveform parameter of the eye pattern or queries the current setting.	5-138
:TELecomtest:EYEPattern:<Parameter>:VALue?	Queries the waveform parameter value of the eye pattern.	5-138
:TELecomtest:EYEPattern:TLEVels?	Queries all settings related to the threshold level of the eye pattern.	5-138
:TELecomtest:EYEPattern:TLEVels:MODE	Sets the unit of the threshold level of the eye pattern or queries the current setting.	5-138
:TELecomtest:EYEPattern:TLEVels:PERCent	Sets the threshold level of the eye pattern as a percentage or queries the current setting.	5-138
:TELecomtest:EYEPattern:TLEVels:UNIT	Sets the threshold level of the eye pattern in UNIT or queries the current setting.	5-138
:TELecomtest:EYEPattern:VDARk	Sets the dark level (zero light level) or queries the current setting.	5-139
:TELecomtest:MASK?	Queries all settings related to the mask test.	5-139
:TELecomtest:MASK:ELEMent<x>?	Queries all settings related to the element used in the mask test.	5-139
:TELecomtest:MASK:ELEMent<x>:ALL	Turns ON/OFF all items of the element.	5-139
:TELecomtest:MASK:ELEMent<x>:PSPCount?	Queries the settings related to the error rate for the number of sampled data points of the element.	5-139
:TELecomtest:MASK:ELEMent<x>:PSPCount:STATe	Turns ON/OFF the measurement of the error rate for the number of sampled data points of the element or queries the current setting.	5-139
:TELecomtest:MASK:ELEMent<x>:PSPCount:VALue?	Queries the error rate for the number of sampled data points of the element.	5-139
:TELecomtest:MASK:ELEMent<x>:PWCount?	Queries the settings related to the error rate for the acquisition count of the element.	5-139
:TELecomtest:MASK:ELEMent<x>:PWCount:STATe	Turns ON/OFF the measurement of the error rate for the acquisition count of the element or queries the current setting.	5-140
:TELecomtest:MASK:ELEMent<x>:PWCount:VALue?	Queries the error rate for the acquisition count of the element.	5-140

Command	Function	Page
:TELecomtest:MASK:ELEMent<x>:SPCount?	Queries the settings related to the number of sampled data points for the element that results in error.	5-140
:TELecomtest:MASK:ELEMent<x>:SPCount:STATe	Turns ON/OFF the measurement of the number of sampled data points for the element that results in error or queries the current setting.	5-140
:TELecomtest:MASK:ELEMent<x>:SPCount:VALue?	Queries the number of sampled data points for the element that resulted in error.	5-140
:TELecomtest:MASK:ELEMent<x>:WCOunt?	Queries the settings related to the acquisition count for the element that results in error.	5-140
:TELecomtest:MASK:ELEMent<x>:WCOunt:STATe	Turns ON/OFF the measurement of the acquisition count for the element that results in error or queries the current setting.	5-140
:TELecomtest:MASK:ELEMent<x>:WCOunt:VALue?	Queries the acquisition count for the element that resulted in error.	5-140
:TELecomtest:MMODe	Turns ON/OFF the mask display or queries the current setting.	5-141
:TELecomtest:TRACe	Sets the source trace of the telecom test or queries the current setting.	5-141
:TELecomtest:TRANge	Sets the measurement range of the telecom test or queries the current setting.	5-141
:TELecomtest:WINDow	Sets the measurement source window of the telecom test or queries the current setting.	5-141
TIMebase Group		
:TIMebase?	Queries all settings related to the time base.	5-142
:TIMebase:SRATe?	Queries the sample rate.	5-142
:TIMebase:TDIV	Sets the T/div value or queries the current setting.	5-142
TRIGger Group		
:TRIGger?	Queries all settings related to the trigger.	5-143
:TRIGger:ACTion?	Queries all settings related to the action-on-trigger.	5-143
:TRIGger:ACTion:ACQCount	Sets the action count of action-on-trigger or queries the current setting.	5-143
:TRIGger:ACTion:BUZZer	Sets whether to sound a buzzer when an action is activated or queries the current setting.	5-143
:TRIGger:ACTion:HCOPy	Sets whether to output screen image data (ON/OFF) when an action is activated or queries the current setting.	5-143
:TRIGger:ACTion:MODE	Sets the action-on-trigger mode or queries the current setting.	5-143
:TRIGger:ACTion:SAVE	Sets whether to save the waveform data to the storage medium (ON/OFF) when an action is activated or queries the current setting.	5-144
:TRIGger:ACTion:STARt	Starts the action-on-trigger.	5-144
:TRIGger:ACTion:STOP	Stops the action-on-trigger.	5-144
:TRIGger:CLOCk?	Queries all settings related to the clock channel.	5-144
:TRIGger:CLOCk:POLarity	Sets the polarity of the clock channel or queries the current setting.	5-144
:TRIGger:CLOCk:SOURce	Sets the source waveform of the clock channel or queries the current setting.	5-144
:TRIGger:DELay?	Queries all settings related to the trigger delay.	5-144
:TRIGger:DELay:EDGecount?	Queries all settings related to edge count of the trigger delay.	5-144
:TRIGger:DELay:EDGecount:COUNt	Sets the edge count value of the trigger delay or queries the current setting.	5-144
:TRIGger:DELay:MODE	Turns ON/OFF the trigger delay or queries the current setting.	5-144
:TRIGger:DELay:POLarity	Sets the edge polarity the trigger delay or queries the current setting.	5-144
:TRIGger:DELay:SOURce	Sets the edge source the trigger delay or queries the current setting.	5-145
:TRIGger:DELay:TIME	Sets the delay value the trigger delay or queries the current setting.	5-145
:TRIGger:DELay:TYPE	Sets the trigger delay type or queries the current setting.	5-145
:TRIGger:EINTerval?	Queries all settings related to the event interval.	5-145
:TRIGger:EINTerval:EVENt<x>?	Queries all settings related to the event.	5-146
:TRIGger:EINTerval:EVENt<x>:CLOCk?	Queries all settings related to the clock channel of the event.	5-146
:TRIGger:EINTerval:EVENt<x>:CLOCk:POLarity	Sets the polarity of the clock channel of the event or queries the current setting.	5-146

Command	Function	Page
:TRIGger:EINTerval:EVENt<x>:CLOCk:SOURce	Sets the source waveform of the clock channel of the event or queries the current setting.	5-146
:TRIGger:EINTerval:EVENt<x>:ESTate?	Queries all settings related to the edge/state trigger.	5-146
:TRIGger:EINTerval:EVENt<x>:ESTate:POLarity	Sets the polarity of the edge/state trigger or queries the current setting.	5-147
:TRIGger:EINTerval:EVENt<x>:ESTate:SOURce	Sets the trigger source of the edge/state trigger or queries the current setting.	5-147
:TRIGger:EINTerval:EVENt<x>:I2CBus?	Queries all settings related to the I^2C bus trigger of the event.	5-147
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa?	Queries all settings related to the address of the I^2C bus trigger.	5-147
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT10address?	Queries all settings related to the 10-bit address of the I^2C bus trigger.	5-148
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT10address:HEXA	Sets the 10-bit address of the I^2C bus trigger in hexadecimal notation.	5-148
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT10address:PATTern	Sets the 10-bit address of the I^2C bus trigger in binary notation or queries the current setting.	5-148
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress?	Queries all settings related to the 7-bit address of the I^2C bus trigger.	5-148
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress:HEXA	Sets the 7-bit address of the I^2C bus trigger in hexadecimal notation.	5-148
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress:PATTern	Sets the 7-bit address of the I^2C bus trigger in binary notation or queries the current setting.	5-148
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub?	Queries all settings related to the 7-bit + Sub address of the I^2C bus trigger.	5-148
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess?	Queries all settings related to the 7-bit address of the 7-bit + Sub address of the I^2C bus trigger.	5-149
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess:HEXA	Sets the 7-bit address of the 7-bit + Sub address of the I^2C bus trigger in hexadecimal notation.	5-149
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess:PATTern	Sets the 7-bit address of the 7-bit + Sub address of the I^2C bus trigger in binary notation or queries the current setting.	5-149
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress?	Queries all settings related to the Sub address of the 7-bit + Sub address of the I^2C bus trigger.	5-149
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress:HEXA	Sets the Sub address of the 7-bit + Sub address of the I^2C bus trigger in hexadecimal notation.	5-149
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress:PATTern	Sets the Sub address of the 7-bit + Sub address of the I^2C bus trigger in binary notation or queries the current setting.	5-149
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:TYPE	Sets the address type of the I^2C bus trigger or queries the current setting.	5-149
:TRIGger:EINTerval:EVENt<x>:I2CBus:CLOCk?	Queries all settings related to the clock of the I^2C bus trigger.	5-150
:TRIGger:EINTerval:EVENt<x>:I2CBus:CLOCk:SOURce	Sets the clock trace of the I^2C bus trigger or queries the current setting.	5-150
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA?	Queries all settings related to the data of the I^2C bus trigger.	5-150
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:BYTE	Sets the number of data bytes of the I^2C bus trigger or queries the current setting.	5-150

Command	Function	Page
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:CONDition	Sets the determination method (match or not match) of the data of the I²C bus trigger or queries the current setting.	5-150
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:DPOSition	Sets the position for comparing the data pattern of the I²C bus trigger or queries the current setting.	5-150
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:HEXA<x>	Sets the data of the I²C bus trigger in hexadecimal notation.	5-151
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:MODE	Enables/Disables the data conditions of the I²C bus trigger or queries the current setting.	5-151
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:PATTern<x>	Sets the data of the I²C bus trigger in binary notation or queries the current setting.	5-151
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:PMODe	Sets the pattern comparison start position mode of the data of the I²C bus trigger or queries the current setting.	5-151
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:SOURce	Sets the data trace of the I²C bus trigger or queries the current setting.	5-151
:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl?	Queries all settings related to the general call of the I²C bus trigger.	5-151
:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:BIT7maddress?	Queries all settings related to the 7-bit master address of the general call of the I²C bus trigger.	5-152
:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:BIT7maddress:HEXA	Sets the 7-bit master address of the general call of the I²C bus trigger in hexadecimal notation.	5-152
:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:BIT7maddress:PATTern	Sets the 7-bit master address of the general call of the I²C bus trigger in binary notation or queries the current setting.	5-152
:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:SBYTe	Sets the second byte type of the general call of the I²C bus trigger or queries the current setting.	5-152
:TRIGger:EINTerval:EVENt<x>:I2CBus:MODE	Sets the trigger mode of the I²C bus trigger or queries the current setting.	5-152
:TRIGger:EINTerval:EVENt<x>:I2CBus:NAIGnore?	Queries all settings related to the NON ACK ignore mode of the I²C bus trigger.	5-152
:TRIGger:EINTerval:EVENt<x>:I2CBus:NAIGnore:HSMode	Sets whether to ignore NON ACK in high speed mode of the I²C bus trigger or queries the current setting.	5-153
:TRIGger:EINTerval:EVENt<x>:I2CBus:NAIGnore:RACCess	Sets whether to ignore NON ACK in read access mode of the I²C bus trigger or queries the current setting.	5-153
:TRIGger:EINTerval:EVENt<x>:I2CBus:NAIGnore:SBYTe	Sets whether to ignore NON ACK in the start byte of the I²C bus trigger or queries the current setting.	5-153
:TRIGger:EINTerval:EVENt<x>:I2CBus:SBHSmode?	Queries all settings related to the start byte and high speed mode of the I²C bus trigger.	5-153
:TRIGger:EINTerval:EVENt<x>:I2CBus:SBHSmode:TYPE	Sets the type of the start byte or high speed mode of the I²C bus trigger or queries the current setting.	5-153
:TRIGger:EINTerval:EVENt<x>:SPATtern?	Queries all settings related to the serial pattern trigger of the event.	5-153
:TRIGger:EINTerval:EVENt<x>:SPATtern:BITRate	Sets the bit rate of the serial pattern trigger or queries the current setting.	5-154

Command	Function	Page
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLEar	Clears the entire pattern of the serial pattern trigger.	5-154
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk?	Queries all settings related to clock of the serial pattern trigger.	5-154
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:MODE	Enables/Disables the clock of the serial pattern trigger or queries the current setting.	5-154
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:POLarity	Sets the polarity of the clock trace of the serial pattern trigger or queries the current setting.	5-154
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:SOURce	Sets the clock trace of the serial pattern trigger or queries the current setting.	5-154
:TRIGger:EINTerval:EVENt<x>:SPATtern:CS	Enables/Disables the chip select of the serial pattern trigger or queries the current setting.	5-154
:TRIGger:EINTerval:EVENt<x>:SPATtern:DATA?	Queries all settings related to data of the serial pattern trigger.	5-155
:TRIGger:EINTerval:EVENt<x>:SPATtern:DATA:ACTive	Sets the active level of the data of the serial pattern trigger or queries the current setting.	5-155
:TRIGger:EINTerval:EVENt<x>:SPATtern:DATA:SOURce	Sets the data trace of the serial pattern trigger or queries the current setting.	5-155
:TRIGger:EINTerval:EVENt<x>:SPATtern:HEXA	Sets the pattern of the serial pattern trigger in hexadecimal notation.	5-155
:TRIGger:EINTerval:EVENt<x>:SPATtern:LATCh?	Queries all settings related to latch of the serial pattern trigger.	5-155
:TRIGger:EINTerval:EVENt<x>:SPATtern:LATCh:POLarity	Sets the polarity of the latch trace of the serial pattern trigger or queries the current setting.	5-155
:TRIGger:EINTerval:EVENt<x>:SPATtern:LATCh:SOURce	Sets the latch trace of the serial pattern trigger or queries the current setting.	5-155
:TRIGger:EINTerval:EVENt<x>:SPATtern:PATTern	Sets the pattern of the serial pattern trigger in binary notation or queries the current setting.	5-156
:TRIGger:EINTerval:EVENt<x>:SPIBus?	Queries all settings related to the SPI bus trigger of the event.	5-156
:TRIGger:EINTerval:EVENt<x>:SPIBus:BITorder	Sets the bit order of the SPI bus trigger or queries the current setting.	5-156
:TRIGger:EINTerval:EVENt<x>:SPIBus:CLOCk?	Queries all settings related to the clock of the SPI bus trigger.	5-156
:TRIGger:EINTerval:EVENt<x>:SPIBus:CLOCk:POLarity	Sets the polarity of the clock trace of the SPI bus trigger or queries the current setting.	5-156
:TRIGger:EINTerval:EVENt<x>:SPIBus:CLOCk:SOURce	Sets the clock trace of the SPI bus trigger or queries the current setting.	5-156
:TRIGger:EINTerval:EVENt<x>:SPIBus:CS?	Queries all settings related to the chip select of the SPI bus trigger.	5-156
:TRIGger:EINTerval:EVENt<x>:SPIBus:CS:ACTive	Sets the active level of the chip select of the SPI bus trigger or queries the current setting.	5-157
:TRIGger:EINTerval:EVENt<x>:SPIBus:CS:SOURce	Sets the chip select trace of the SPI bus trigger or queries the current setting.	5-157
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>?	Queries all settings related to the data of the SPI bus trigger.	5-157
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:BYTE	Sets the number of bytes of the data of the SPI bus trigger or queries the current setting.	5-157

Command	Function	Page
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:CONDition	Sets the determination method (match or not match) of the data of the SPI bus trigger or queries the current setting.	5-157
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:DPOSition	Sets the pattern comparison start position of the data of the SPI bus trigger or queries the current setting.	5-157
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:HEXA<x>	Sets the data of the SPI bus trigger in hexadecimal notation.	5-158
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:PATTern<x>	Sets the data of the SPI bus trigger in binary notation or queries the current setting.	5-158
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:SOURce	Sets the trace of the data of the SPI bus trigger or queries the current setting.	5-158
:TRIGger:EINTerval:EVENt<x>:SPIBus:MODE	Sets the wiring system of the SPI bus trigger (three-wire or four-wire) or queries the current setting.	5-158
:TRIGger:EINTerval:EVENt<x>:STATe?	Queries all settings related to the state trigger of the event.	5-158
:TRIGger:EINTerval:EVENt<x>:STATe:CHANnel<x>	Sets the condition to be satisfied of the channel or queries the current setting.	5-158
:TRIGger:EINTerval:EVENt<x>:STATe:LOGic	Sets the logic of the condition to be satisfied or queries the current setting.	5-159
:TRIGger:EINTerval:EVENt<x>:TYPE	Sets the trigger type of the event or queries the current setting.	5-159
:TRIGger:EINTerval:EVENt<x>:WIDTh?	Queries all settings related to the pulse width trigger of the event.	5-159
:TRIGger:EINTerval:EVENt<x>:WIDTh:MODE	Sets the determination mode of the pulse width trigger or queries the current setting.	5-159
:TRIGger:EINTerval:EVENt<x>:WIDTh:POLarity	Sets the polarity of the pulse width trigger or queries the current setting.	5-159
:TRIGger:EINTerval:EVENt<x>:WIDTh:SOURce	Sets the trigger source of the pulse width trigger or queries the current setting.	5-160
:TRIGger:EINTerval:EVENt<x>:WIDTh:TIME<x>	Sets the pulse width of the pulse width trigger or queries the current setting.	5-160
:TRIGger:EINTerval:MODE	Sets the determination mode of the event interval or queries the current setting.	5-160
:TRIGger:EINTerval:TIME<x>	Sets the interval time of the event interval or queries the current setting.	5-160
:TRIGger:EINTerval:TRY?	Queries all settings related to the event interval trial.	5-160
:TRIGger:EINTerval:TRY:MODE	Sets the trial mode or queries the current setting.	5-160
:TRIGger:EINTerval:TRY:SELect	Sets the source event of the trial mode or queries the current setting.	5-160
:TRIGger:ENHanced?	Queries all settings related to the enhanced trigger.	5-161
:TRIGger:ENHanced:I2CBus?	Queries all settings related to the I^2C bus trigger.	5-161
:TRIGger:ENHanced:I2CBus:ADATa?	Queries all settings related to the address of the I^2C bus trigger.	5-161
:TRIGger:ENHanced:I2CBus:ADATa:BIT10address?	Queries all settings related to the 10-bit address of the I^2C bus trigger.	5-161
:TRIGger:ENHanced:I2CBus:ADATa:BIT10address:HEXA	Sets the 10-bit address of the I^2C bus trigger in hexadecimal notation.	5-162
:TRIGger:ENHanced:I2CBus:ADATa:BIT10address:PATTern	Sets the 10-bit address of the I^2C bus trigger in binary notation or queries the current setting.	5-162
:TRIGger:ENHanced:I2CBus:ADATa:BIT7ADdress?	Queries all settings related to the 7-bit address of the I^2C bus trigger.	5-162
:TRIGger:ENHanced:I2CBus:ADATa:BIT7ADdress:HEXA	Sets the 7-bit address of the I^2C bus trigger in hexadecimal notation.	5-162
:TRIGger:ENHanced:I2CBus:ADATa:BIT7ADdress:PATTern	Sets the 7-bit address of the I^2C bus trigger in binary notation or queries the current setting.	5-162
:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub?	Queries all settings related to the 7-bit + Sub address of the I^2C bus trigger.	5-162

Command	Function	Page
:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:ADDRess?	Queries all settings related to the 7-bit address of the 7-bit + Sub address of the I²C bus trigger.	5-162
:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:ADDRess:HEXA	Sets the 7-bit address of the 7-bit + Sub address of the I²C bus trigger in hexadecimal notation.	5-162
:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:ADDRess:PATTern	Sets the 7-bit address of the 7-bit + Sub address of the I²C bus trigger in binary notation or queries the current setting.	5-163
:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:SADDress?	Queries all settings related to the Sub address of the 7-bit + Sub address of the I²C bus trigger.	5-163
:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:SADDress:HEXA	Sets the Sub address of the 7-bit + Sub address of the I²C bus trigger in hexadecimal notation.	5-163
:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:SADDress:PATTern	Sets the Sub address of the 7-bit + Sub address of the I²C bus trigger in binary notation or queries the current setting.	5-163
:TRIGger:ENHanced:I2CBus:ADATa:TYPE	Sets the address type of the I2C bus trigger or queries the current setting.	5-163
:TRIGger:ENHanced:I2CBus:CLOCk?	Queries all settings related to the clock of the I²C bus trigger.	5-163
:TRIGger:ENHanced:I2CBus:CLOCk:SOURce	Sets the clock trace of the I²C bus trigger or queries the current setting.	5-163
:TRIGger:ENHanced:I2CBus:DATA?	Queries all settings related to the data of the I²C bus trigger.	5-163
:TRIGger:ENHanced:I2CBus:DATA:BYTE	Sets the number of data bytes of the I²C bus trigger or queries the current setting.	5-164
:TRIGger:ENHanced:I2CBus:DATA:CONDition	Sets the determination method (match or not match) of the data of the I²C bus trigger or queries the current setting.	5-164
:TRIGger:ENHanced:I2CBus:DATA:DPOSition	Sets the position for comparing the data pattern of the I²C bus trigger or queries the current setting.	5-164
:TRIGger:ENHanced:I2CBus:DATA:HEXA<x>	Sets the data of the I²C bus trigger in hexadecimal notation.	5-164
:TRIGger:ENHanced:I2CBus:DATA:MODE	Enables/Disables the data conditions of the I²C bus trigger or queries the current setting.	5-164
:TRIGger:ENHanced:I2CBus:DATA:PATTern<x>	Sets the data of the I²C bus trigger in binary notation or queries the current setting.	5-164
:TRIGger:ENHanced:I2CBus:DATA:PMODe	Sets the pattern comparison start position mode of the data of the I²C bus trigger or queries the current setting.	5-164
:TRIGger:ENHanced:I2CBus:DATA:SOURce	Sets the data trace of the I²C bus trigger or queries the current setting.	5-164
:TRIGger:ENHanced:I2CBus:GCALl?	Queries all settings related to the general call of the I²C bus trigger.	5-165
:TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress?	Queries all settings related to the 7-bit master address of the general call of the I²C bus trigger.	5-165
:TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress:HEXA	Sets the 7-bit master address of the general call of the I²C bus trigger in hexadecimal notation.	5-165
:TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress:PATTern	Sets the 7-bit master address of the general call of the I²C bus trigger in binary notation or queries the current setting.	5-165
:TRIGger:ENHanced:I2CBus:GCALl:SBYTe	Sets the second byte type of the general call of the I²C bus trigger or queries the current setting.	5-165

Command	Function	Page
:TRIGger:ENHanced:I2CBus:MODE	Sets the trigger mode of the I²C bus trigger or queries the current setting.	5-165
:TRIGger:ENHanced:I2CBus:NAIGnore?	Queries all settings related to the NON ACK ignore mode of the I²C bus trigger.	5-165
:TRIGger:ENHanced:I2CBus:NAIGnore:HSMode	Sets whether to ignore NON ACK in high speed mode of the I²C bus trigger or queries the current setting.	5-165
:TRIGger:ENHanced:I2CBus:NAIGnore:RACCess	Sets whether to ignore NON ACK in read access mode of the I²C bus trigger or queries the current setting.	5-166
:TRIGger:ENHanced:I2CBus:NAIGnore:SBYTe	Sets whether to ignore NON ACK in the start byte of the I²C bus trigger or queries the current setting.	5-166
:TRIGger:ENHanced:I2CBus:SBHSmode?	Queries all settings related to the start byte and high speed mode of the I²C bus trigger.	5-166
:TRIGger:ENHanced:I2CBus:SBHSmode:TYPE	Sets the type of the start byte or high speed mode of the I²C bus trigger or queries the current setting.	5-166
:TRIGger:ENHanced:SPATtern?	Queries all settings related to the serial pattern trigger.	5-166
:TRIGger:ENHanced:SPATtern:BITRate	Sets the bit rate of the serial pattern trigger or queries the current setting.	5-166
:TRIGger:ENHanced:SPATtern:CLEar	Clears the entire pattern of the serial pattern trigger.	5-166
:TRIGger:ENHanced:SPATtern:CLOCk?	Queries all settings related to clock of the serial pattern trigger.	5-166
:TRIGger:ENHanced:SPATtern:CLOCk:MODE	Enables/Disables the clock of the serial pattern trigger or queries the current setting.	5-166
:TRIGger:ENHanced:SPATtern:CLOCk:POLarity	Sets the polarity of the clock trace of the serial pattern trigger or queries the current setting.	5-167
:TRIGger:ENHanced:SPATtern:CLOCk:SOURce	Sets the clock trace of the serial pattern trigger or queries the current setting.	5-167
:TRIGger:ENHanced:SPATtern:CS	Enables/Disables the chip select of the serial pattern trigger or queries the current setting.	5-167
:TRIGger:ENHanced:SPATtern:DATA?	Queries all settings related to data of the serial pattern trigger.	5-167
:TRIGger:ENHanced:SPATtern:DATA:ACTive	Sets the active level of the data of the serial pattern trigger or queries the current setting.	5-167
:TRIGger:ENHanced:SPATtern:DATA:SOURce	Sets the data trace of the serial pattern trigger or queries the current setting.	5-167
:TRIGger:ENHanced:SPATtern:HEXA	Sets the pattern of the serial pattern trigger in hexadecimal notation.	5-167
:TRIGger:ENHanced:SPATtern:LATCh?	Queries all settings related to latch of the serial pattern trigger.	5-167
:TRIGger:ENHanced:SPATtern:LATCh:POLarity	Sets the polarity of the latch trace of the serial pattern trigger or queries the current setting.	5-168
:TRIGger:ENHanced:SPATtern:LATCh:SOURce	Sets the latch trace of the serial pattern trigger or queries the current setting.	5-168
:TRIGger:ENHanced:SPATtern:PATTern	Sets the pattern of the serial pattern trigger in binary notation or queries the current setting.	5-168
:TRIGger:ENHanced:SPIBus?	Queries all settings related to the SPI bus trigger.	5-168
:TRIGger:ENHanced:SPIBus:BITorder	Sets the bit order of the SPI bus trigger or queries the current setting.	5-168
:TRIGger:ENHanced:SPIBus:CLOCk?	Queries all settings related to the clock of the SPI bus trigger.	5-168
:TRIGger:ENHanced:SPIBus:CLOCk:POLarity	Sets the polarity of the clock trace of the SPI bus trigger or queries the current setting.	5-168
:TRIGger:ENHanced:SPIBus:CLOCk:SOURce	Sets the clock trace of the SPI bus trigger or queries the current setting.	5-169
:TRIGger:ENHanced:SPIBus:CS?	Queries all settings related to the chip select of the SPI bus trigger.	5-169
:TRIGger:ENHanced:SPIBus:CS:ACTive	Sets the active level of the chip select of the SPI bus trigger or queries the current setting.	5-169

Command	Function	Page
:TRIGger:ENHanced:SPIBus:CS:SOURce	Sets the chip select trace of the SPI bus trigger or queries the current setting.	5-169
:TRIGger:ENHanced:SPIBus:DATA<x>?	Queries all settings related to the data of the SPI bus trigger.	5-169
:TRIGger:ENHanced:SPIBus:DATA<x>:BYTE	Sets the number of bytes of the data of the SPI bus trigger or queries the current setting.	5-169
:TRIGger:ENHanced:SPIBus:DATA<x>:CONDition	Sets the determination method (match or not match) of the data of the SPI bus trigger or queries the current setting.	5-169
:TRIGger:ENHanced:SPIBus:DATA<x>:DPOSition	Sets the pattern comparison start position of the data of the SPI bus trigger or queries the current setting.	5-169
:TRIGger:ENHanced:SPIBus:DATA<x>:HEXA<x>	Sets the data of the SPI bus trigger in hexadecimal notation.	5-170
:TRIGger:ENHanced:SPIBus:DATA<x>:PATTern<x>	Sets the data of the SPI bus trigger in binary notation or queries the current setting.	5-170
:TRIGger:ENHanced:SPIBus:DATA<x>:SOURce	Sets the trace of the data of the SPI bus trigger or queries the current setting.	5-170
:TRIGger:ENHanced:SPIBus:MODE	Sets the wiring system of the SPI bus trigger (three-wire or four-wire) or queries the current setting.	5-170
:TRIGger:ENHanced:TV?	Queries all settings related to the TV trigger.	5-170
:TRIGger:ENHanced:TV:COUPling?	Queries the trigger coupling of the TV trigger.	5-170
:TRIGger:ENHanced:TV:CUSTomize	Turns ON/OFF the sync guard function of the TV trigger or queries the current setting.	5-170
:TRIGger:ENHanced:TV:FIELd	Sets the field of the TV trigger or queries the current setting.	5-170
:TRIGger:ENHanced:TV:FRAMe	Sets the frame skip function of the TV trigger or queries the current setting.	5-170
:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|USERdefine}?	Queries all settings related to the TV trigger mode.	5-171
:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL}:HFRejection?	Queries the low pass filter (HF rejection) of the TV trigger.	5-171
:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|USERdefine}:LINE	Sets the line for activating the TV trigger or queries the current setting.	5-171
:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|USERdefine}:POLarity	Sets the input polarity of the TV trigger or queries the current setting.	5-171
:TRIGger:ENHanced:TV:LEVel	Sets the trigger level of the TV trigger or queries the current setting.	5-171
:TRIGger:ENHanced:TV:SGUard	Sets the sync guard of the TV trigger or queries the current setting.	5-171
:TRIGger:ENHanced:TV:SOURce	Sets the trigger source of the TV trigger or queries the current setting.	5-171
:TRIGger:ENHanced:TV:TYPE	Sets the input type of the TV trigger or queries the current setting.	5-171
:TRIGger:ENHanced:TV:USERdefine:DEFinition	Sets the user defined resolution or queries the current setting.	5-171
:TRIGger:ENHanced:TV:USERdefine:HFRejection	Sets the user-defined low pass filter (HF rejection) or queries the current setting.	5-172
:TRIGger:ENHanced:TV:USERdefine:HSYNc	Sets the user-defined horizontal sync signal or queries the current setting.	5-172
:TRIGger:ESTate?	Queries all settings related to the edge/state trigger.	5-172
:TRIGger:ESTate:EOR?	Queries all settings related to the OR trigger.	5-172
:TRIGger:ESTate:EOR:CHANnel<x>	Sets the channel polarity of the OR trigger or queries the current setting.	5-172
:TRIGger:ESTate:POLarity	Sets the polarity of the edge/state trigger or queries the current setting.	5-172
:TRIGger:ESTate:SOURce	Sets the trigger source of the edge/state trigger or queries the current setting.	5-172
:TRIGger:HOLDoff	Sets the hold off time or queries the current setting.	5-173
:TRIGger:MODE	Sets the trigger mode or queries the current setting.	5-173
:TRIGger:POSition	Sets the trigger position or queries the current setting.	5-173
: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.	5-173
:TRIGger:SOURce?	Queries all settings related to the trigger source.	5-173

Command	Function	Page
:TRIGger:SOURce:CHANnel<x>?	Queries all settings related to the channel of the trigger source.	5-173
:TRIGger:SOURce:CHANnel<x>:COUPling	Sets the trigger coupling of the channel or queries the current setting.	5-173
:TRIGger:SOURce:CHANnel<x>:HFRejection	Sets the low pass filter (HF rejection) of the channel or queries the current setting.	5-174
:TRIGger:SOURce:CHANnel<x>:HYSTeresis	Sets the hysteresis of the channel or queries the current setting.	5-174
:TRIGger:SOURce:CHANnel<x>:LEVel	Sets the trigger level of the channel or queries the current setting.	5-174
:TRIGger:SOURce:CHANnel<x>:STATe	Sets the condition to be satisfied of the channel or queries the current setting.	5-174
:TRIGger:SOURce:CHANnel<x>:WIDTh	Sets the window trigger width of the channel or queries the current setting.	5-174
:TRIGger:SOURce:CHANnel<x>:WINDow	Turns ON/OFF the window of the channel or queries the current setting.	5-174
:TRIGger:SOURce:EXTernal?	Queries all settings related to the external trigger.	5-174
:TRIGger:SOURce:EXTernal:LEVel	Sets the trigger level of the external trigger or queries the current setting.	5-175
:TRIGger:SOURce:EXTernal:PROBe	Sets the probe attenuation of the external trigger or queries the current setting.	5-175
:TRIGger:SOURce:LOGic	Sets the trigger source logic or queries the current setting.	5-175
:TRIGger:TYPE	Sets the trigger type or queries the current setting.	5-175
:TRIGger:WIDTh?	Queries all settings related to the pulse width trigger.	5-175
:TRIGger:WIDTh:MODE	Sets the determination mode of the pulse width trigger or queries the current setting.	5-175
:TRIGger:WIDTh:POLarity	Sets the polarity of the pulse width trigger or queries the current setting.	5-175
:TRIGger:WIDTh:SOURce	Sets the trigger source of the pulse width trigger or queries the current setting.	5-175
:TRIGger:WIDTh:TIME<x>	Sets the pulse width of the pulse width trigger or queries the current setting.	5-175
WAVeform Group		
:WAVeform?	Queries all of the information of the waveform data.	5-176
:WAVeform:BITS?	Queries the bit length of the specified waveform data.	5-176
:WAVeform:BYTeorder	Sets the transmission byte order or queries the current setting.	5-176
:WAVeform:END	Sets the last data point of the specified waveform or queries the current setting.	5-176
:WAVeform:FORMat	Sets the format of the data to be transmitted or queries the current setting.	5-176
:WAVeform:LENGth?	Queries the total number of data points of the specified waveform.	5-176
:WAVeform:OFFSet?	Queries the offset value of the specified waveform data.	5-176
:WAVeform:RANGe?	Queries the range value of the specified waveform data.	5-176
:WAVeform:RECord	Sets the target record number for the commands in the WAVeform group or queries the current setting.	5-176
:WAVeform:RECord? MINimum	Queries the minimum record number.	5-176
:WAVeform:SEND?	Queries the specified waveform data.	5-177
:WAVeform:SIGN?	Queries the presence of a sign.	5-177
:WAVeform:SRATe?	Queries the sample rate of the target record.	5-177
:WAVeform:STARt	Sets the first data point of the specified waveform or queries the current setting.	5-177
:WAVeform:TRACe	Sets the target waveform for the commands in the WAVeform group or queries the current setting.	5-177
:WAVeform:TRIGger?	Queries the trigger position of the target record.	5-177
:WAVeform:TYPE?	Queries the acquisition mode of the specified waveform.	5-177
ZOOM Group		
:ZOOM?	Queries all settings related to the waveform zoom.	5-178
:ZOOM:ALLocation?	Queries all settings related to the zoom source waveform.	5-178
:ZOOM:ALLocation:ALLon	Sets all waveforms to be zoomed.	5-178
:ZOOM:ALLocation:TRACe<x>	Turns ON/OFF the trace you wish to zoom or queries the current setting.	5-178
:ZOOM:FORMat<x>	Sets the display format of the zoom waveform or queries the current setting.	5-178
:ZOOM:HLINkage	Turns ON/OFF the horizontal link or queries the current setting.	5-178
:ZOOM:HORizontal<x>?	Queries all settings related to the horizontal zoom.	5-178
:ZOOM:HORizontal<x>:ASCRoll?	Queries all settings related to the auto scroll function.	5-178
:ZOOM:HORizontal<x>:ASCRoll:JUMP	Moves the zoom center position to the left or right edge.	5-178
:ZOOM:HORizontal<x>:ASCRoll:SPEed	Sets the auto scroll speed or queries the current setting.	5-178
:ZOOM:HORizontal<x>:ASCRoll:STARt	Starts auto scrolling.	5-179
:ZOOM:HORizontal<x>:ASCRoll:STOP	Stops auto scrolling.	5-179

Command	Function	Page
:ZOOM:HORizontal<x>:MAG	Sets the horizontal zoom magnification or queries the current setting.	5-179
:ZOOM:HORizontal<x>:POSition	Sets the horizontal zoom center position or queries the current setting.	5-179
:ZOOM:MODE	Sets the display mode of the zoom waveform or queries the current setting.	5-179
:ZOOM:TYPE<x>	Sets the zoom type or queries the current setting.	5-179
:ZOOM:VERTical<x>?	Queries all settings related to the vertical zoom.	5-179
:ZOOM:VERTical<x>:INITialize	Initializes the vertical zoom.	5-179
:ZOOM:VERTical<x>:MAG	Sets the vertical zoom magnification or queries the current setting.	5-179
:ZOOM:VERTical<x>:POSition	Sets the vertical zoom position or queries the current setting.	5-179
:ZOOM:VERTical<x>:TRACe	Sets the displayed trace of the vertical zoom screen or queries the current setting.	5-179
:ZOOM:VLINkage	Turns ON/OFF the vertical link or queries the current setting.	5-179
Common Command Group		
*CAL?	Performs calibration and queries the result.	5-180
*CLS	Clears the standard event register, extended event register, and error queue.	5-180
*ESE	Sets the standard event enable register or queries the current setting.	5-180
*ESR?	Queries the standard event register and clears the register.	5-180
*IDN?	Queries the instrument model.	5-180
*LRN?	Queries collectively the current settings of the command group.	5-181
*OPC	Sets the OPC bit to 1 after the completion of the specified overlap command.	5-182
*OPC?	Creates a response after the completion of the specified overlap command.	5-182
*OPT?	Queries the options.	5-182
*PSC	Sets whether to clear the registers at power on or queries the current setting.	5-182
*RST	Executes the initialization of settings.	5-182
*SRE	Sets the service request enable register or queries the current setting.	5-182
*STB?	Queries the status byte register.	5-182
*TST?	Performs a self-test and queries the result.	5-183
*WAI	Holds the subsequent command until the completion of the specified overlap operation.	5-183

5.2 ACQuire Group

:ACQuire?

Function Queries all settings related to the waveform acquisition.

Syntax `:ACQuire?`

Example
```
:ACQUIRE? -> :ACQUIRE:AVERAGE:COUNT 2;
EWEIGHT 16;:ACQUIRE:HRMODE 0;
INTERLEAVE 0;INTERPOLATE 1;MODE NORMAL;
REPETITIVE 0;RLENGTH 12500
```

:ACQuire:AVERage?

Function Queries all settings related to averaging and the waveform acquisition count.

Syntax `:ACQuire:AVERage?`

Example
```
:ACQUIRE:AVERAGE? ->
:ACQUIRE:AVERAGE:COUNT 2;EWEIGHT 16
```

:ACQuire:AVERage:COUNt

Function Sets the waveform acquisition count of averaging mode or queries the current setting.

Syntax
```
:ACQuire:AVERage:COUNt {<NRf>}
:ACQuire:AVERage:COUNt?
```
<NRf> = 2 to 65536 (2^n steps)

Example
```
:ACQUIRE:AVERAGE:COUNT 2
:ACQUIRE:AVERAGE:COUNT? ->
:ACQUIRE:AVERAGE:COUNT 2
```

:ACQuire:AVERage:EWEight

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

Syntax
```
:ACQuire:AVERage:EWEight {<NRf>}
:ACQuire:AVERage:EWEight?
```
<NRf> = 2 to 1024 (2^n steps)

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

:ACQuire:HRMode

Function Turns ON/OFF the high resolution mode or queries the current setting.

Syntax
```
:ACQuire:HRMode {<Boolean>}
:ACQuire:HRMode?
```

Example
```
:ACQUIRE:HRMODE ON
:ACQUIRE:HRMODE? -> :ACQUIRE:HRMODE 1
```

:ACQuire:INTERLeave

Function Turns ON/OFF interleave or queries the current setting.

Syntax
```
:ACQuire:INTERLeave {<Boolean>}
:ACQuire:INTERLeave?
```

Example
```
:ACQUIRE:INTERLEAVE ON
:ACQUIRE:INTERLEAVE? ->
:ACQUIRE:INTERLEAVE 1
```

:ACQuire:INTERPolate

Function Turns ON/OFF data interpolation or queries the current setting.

Syntax
```
:ACQuire:INTERPolate {<Boolean>}
:ACQuire:INTERPolate?
```

Example
```
:ACQUIRE:INTERPOLATE ON
:ACQUIRE:INTERPOLATE? ->
:ACQUIRE:INTERPOLATE 1
```

:ACQuire:MODE

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

Syntax
```
:ACQuire:MODE {AVERage|ENVelope|NORMal}
:ACQuire:MODE?
```

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

:ACQuire:REPetitive

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

Syntax
```
:ACQuire:REPetitive {<Boolean>}
:ACQuire:REPetitive?
```

Example
```
:ACQUIRE:REPETITIVE ON
:ACQUIRE:REPETITIVE? ->
:ACQUIRE:REPETITIVE 1
```

:ACQuire:RLENgth

Function Sets the record length or queries the current setting.

Syntax
```
:ACQuire:RLENgth {<NRf>}
:ACQuire:RLENgth?
```
<NRf> = See the DL9000 User's Manual.

Example
```
:ACQUIRE:RLENGTH 6250000
:ACQUIRE:RLENGTH? ->
:ACQUIRE:RLENGTH 6250000
```

5.3 ANALysis Group

:ANALysis?

Function Queries all settings related to the analysis function.

Syntax :ANALysis?

Example :ANALYSIS? -> :ANALYSIS:
AHISTOGRAM1:HORIZONTAL 0.000E+00,
0.000E+00;MEASURE:CURSOR:BASIC:C1:
STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:
CURSOR:BASIC:C2:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:CURSOR:BASIC:DC:
STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:
CURSOR:CALCULATION:DEFINE1 "C1";
DEFINE2 "C2";DEFINE3 "C1";DEFINE4 "C2";
STATE1 1;STATE2 1;STATE3 1;STATE4 1;:
ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
HLINKAGE 1;HPOSITION1 0.000E+00;
HPOSITION2 1.000E+00;VLINKAGE 1;
VPOSITION1 1.000E+00;
VPOSITION2 1.000E+00;:ANALYSIS:
AHISTOGRAM1:MEASURE:MODE CURSOR;
PARAMETER:AREA1:MAXIMUM:STATE 1;:
ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:
AREA1:MEAN:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:PARAMETER:AREA1:
MEDIAN:STATE 1;:ANALYSIS:AHISTOGRAM1:
MEASURE:PARAMETER:AREA1:MINIMUM:STATE
1;:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:PEAK:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:PARAMETER:AREA1:
RMS:STATE 1;:ANALYSIS:AHISTOGRAM1:
MEASURE:PARAMETER:AREA1:SD2INTEG:
STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:SD3INTEG:STATE 1;:
ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:
AREA1:SDEVIATION:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:PARAMETER:AREA1:
SDINTEG:STATE 1.....

:ANALysis:AHIStogram<x>?

Function Queries all settings related to the accumulated histogram function.

Syntax :ANALysis:AHIStogram<x>?
<x> = 1 or 2

Example :ANALYSIS:AHISTOGRAM1? -> :ANALYSIS:
AHISTOGRAM1:HORIZONTAL 0.000E+00,
0.000E+00;MEASURE:CURSOR:BASIC:C1:
STATE 1;:ANALYSIS:AHISTOGRAM1:
MEASURE:CURSOR:BASIC:C2:STATE 1;:
ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:DC:STATE 1;:ANALYSIS:AHISTOGRAM1:
MEASURE:CURSOR:CALCULATION:
DEFINE1 "C1";DEFINE2 "C2";DEFINE3 "C1";
DEFINE4 "C2";STATE1 1;STATE2 1;
STATE3 1;STATE4 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:CURSOR:HLINKAGE 1;
HPOSITION1 0.000E+00;
HPOSITION2 1.000E+00;VLINKAGE 1;
VPOSITION1 1.000E+00;
VPOSITION2 1.000E+00;:ANALYSIS:
AHISTOGRAM1:MEASURE:MODE CURSOR;
PARAMETER:AREA1:MAXIMUM:STATE 1;:
ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:
AREA1:MEAN:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:PARAMETER:AREA1:
MEDIAN:STATE 1;:ANALYSIS:AHISTOGRAM1:
MEASURE:PARAMETER:AREA1:MINIMUM:
STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:PEAK:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:PARAMETER:AREA1:
RMS:STATE 1;:ANALYSIS:AHISTOGRAM1:
MEASURE:PARAMETER:AREA1:SD2INTEG:
STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:SD3INTEG:STATE 1;:A
NALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:
AREA1:SDEVIATION:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:PARAMETER:AREA1:
SDINTEG:STATE 1.....

:ANALysis:AHIStogram<x>:HORizontal

Function Sets the horizontal range of the accumulated histogram or queries the current setting.

Syntax :ANALysis:AHIStogram<x>:HORizontal
{<NRf>,<NRf>}
:ANALysis:AHIStogram<x>:HORizontal?
<x> = 1 or 2
<NRf> = –4 to 4 (div)

Example :ANALYSIS:AHISTOGRAM1:HORIZONTAL 0,1
:ANALYSIS:AHISTOGRAM1:HORIZONTAL? ->
:ANALYSIS:AHISTOGRAM1:
HORIZONTAL 0.000E+00,0.000E+00

:ANALysis:AHIStogram<x>:MEASure?

:ANALysis:AHIStogram<x>:MEASure?

Function Queries all settings related automated measurement of the accumulated histogram.

Syntax `:ANALysis:AHIStogram<x>:MEASure?`
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:C1:STATE 1;:ANALYSIS:AHISTOGRAM1:
MEASURE:CURSOR:BASIC:C2:STATE 1;:
ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:DC:STATE 1;:ANALYSIS:AHISTOGRAM1:
MEASURE:CURSOR:CALCULATION:
DEFINE1 "C1";DEFINE2 "C2";DEFINE3 "C1";
DEFINE4 "C2";STATE1 1;STATE2 1;
STATE3 1;STATE4 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:CURSOR:HLINKAGE 1;
HPOSITION1 0.000E+00;
HPOSITION2 1.000E+00;VLINKAGE 1;
VPOSITION1 0.000E+00;
VPOSITION2 1.000E+00;:ANALYSIS:
AHISTOGRAM1:MEASURE:MODE CURSOR;
PARAMETER:AREA1:MAXIMUM:STATE 1;:
ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:
AREA1:MEAN:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:PARAMETER:AREA1:
MEDIAN:STATE 1;:ANALYSIS:AHISTOGRAM1:
MEASURE:PARAMETER:AREA1:MINIMUM:
STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:PEAK:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:PARAMETER:AREA1:
RMS:STATE 1;:ANALYSIS:AHISTOGRAM1:
MEASURE:PARAMETER:AREA1:SD2INTEG:
STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:SD3INTEG:STATE 1;:
ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:
AREA1:SDEVIATION:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:PARAMETER:AREA1:
SDINTEG:STATE 1.....
```

:ANALysis:AHIStogram<x>:MEASure:CURSor?

Function Queries all settings related to cursor measurements of the accumulated histogram.

Syntax `:ANALysis:AHIStogram<x>:MEASure:CURSor?`
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR?
-> :ANALYSIS:AHISTOGRAM1:MEASURE:
CURSOR:BASIC:C1:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:CURSOR:BASIC:C2:
STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:
CURSOR:BASIC:DC:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:CURSOR:CALCULATION:
DEFINE1 "C1";DEFINE2 "C2";DEFINE3 "C1";
DEFINE4 "C2";STATE1 1;STATE2 1;
STATE3 1;STATE4 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:CURSOR:HLINKAGE 1;
HPOSITION1 0.000E+00;
HPOSITION2 1.000E+00;VLINKAGE 1;
VPOSITION1 0.000E+00;
VPOSITION2 1.000E+00
```

:ANALysis:AHIStogram<x>:MEASure:CURSor[:BASic]?

Function Queries all settings related to basic items of the cursor of the accumulated histogram.

Syntax `:ANALysis:AHIStogram<x>:MEASure:CURSor[:BASic]?`
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC? -> :ANALYSIS:AHISTOGRAM1:
MEASURE:CURSOR:BASIC:C1:STATE 1;:
ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:C2:STATE 1;:ANALYSIS:AHISTOGRAM1:
MEASURE:CURSOR:BASIC:DC:STATE 1
```

:ANALysis:AHIStogram<x>:MEASure:CURSor[:BASic]:ALL

Function Turns ON/OFF all basic items of the cursor of the accumulated histogram.

Syntax `:ANALysis:AHIStogram<x>:MEASure:CURSor[:BASic]:ALL {<Boolean>}`
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:ALL ON
```

:ANALysis:AHIStogram<x>:MEASure:CURSor[:BASic]:C<x>?

Function Queries all settings related to the cursor of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
CURSor[:BASic]:C<x>?
```
<x> of AHIStogram<x> = 1 or 2
<x> of C<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:C1? -> :ANALYSIS:AHISTOGRAM1:
MEASURE:CURSOR:BASIC:C1:STATE 1
```

:ANALysis:AHIStogram<x>:MEASure:CURSor[:BASic]:C<x>:STATe

Function Turns ON/OFF the cursor of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
CURSor[:BASic]:C<x>:STATe {<Boolean>}
:ANALysis:AHIStogram<x>:MEASure:
CURSor[:BASic]:C<x>:STATe?
```
<x> of AHIStogram<x> = 1 or 2
<x> of C<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:C1:STATE ON
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:C1:STATE? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:C1:STATE 1
```

:ANALysis:AHIStogram<x>:MEASure:CURSor[:BASic]:C<x>:VALue?

Function Queries the measured value of the cursor of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
CURSor[:BASic]:C<x>:VALue?
```
<x> of AHIStogram<x> = 1 or 2
<x> of C<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:C1:VALUE? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:C1:VALUE 1.000E+00
```

:ANALysis:AHIStogram<x>:MEASure:CURSor[:BASic]:DC?

Function Queries all settings related to measured values between cursors of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
CURSor[:BASic]:DC?
```
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:DC? -> :ANALYSIS:AHISTOGRAM1:
MEASURE:CURSOR:BASIC:DC:STATE 1
```

:ANALysis:AHIStogram<x>:MEASure:CURSor[:BASic]:DC:STATe

Function Turns ON/OFF the measured values between cursors of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
CURSor[:BASic]:DC:STATe {<Boolean>}
:ANALysis:AHIStogram<x>:MEASure:
CURSor[:BASic]:DC:STATe?
```
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:DC:STATE ON
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:DC:STATE? ->
:ANALYSIS:AHISTOGRAM1:
MEASURE:CURSOR:BASIC:DC:STATE 1
```

:ANALysis:AHIStogram<x>:MEASure:CURSor[:BASic]:DC:VALue?

Function Queries the measured value between cursors of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
CURSor[:BASic]:DC:VALue?
```
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:DC:VALUE? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
BASIC:DC:VALUE 1.000E+00
```

:ANALysis:AHIStogram<x>:MEASure:CURSor:CALCulation?

Function Queries all settings related to calculation items of the cursor of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:CURSor:
CALCulation?
```
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
CALCULATION? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
CALCULATION:DEFINE1 “C1”;DEFINE2 “C2”;
DEFINE3 “C1”;DEFINE4 “C2”;STATE1 1;
STATE2 1;STATE3 1;STATE4 1
```

:ANALysis:AHIStogram<x>:MEASure:CURSor:CALCulation:ALL

Function Turns ON/OFF all calculation items of the cursor of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:CURSor:
CALCulation:ALL {<Boolean>}
```
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:
CALCULATION:ALL ON
```

:ANALysis:AHIStogram<x>:MEASure:CURSor:CALCulation:DEFine<x>

Function Sets the equation of the calculation item of the cursor of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:CURSor:CALCulation:DEFine<x> {<String>}
:ANALysis:AHIStogram<x>:MEASure:CURSor:CALCulation:DEFine<x>?
```
<x> of AHIStogram<x> = 1 or 2
<x> of DEFine<x> = 1 to 4
<String> = Up to 128 characters

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:CALCULATION:DEFINE1 "C1"
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:CALCULATION:DEFINE1? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:CALCULATION:DEFINE1 "C1"
```

:ANALysis:AHIStogram<x>:MEASure:CURSor:CALCulation:STATe<x>

Function Turns ON/OFF the calculation item of the cursor of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:CURSor:CALCulation:STATe<x> {<Boolean>}
:ANALysis:AHIStogram<x>:MEASure:CURSor:CALCulation:STATe<x>?
```
<x> of AHIStogram<x> = 1 or 2
<x> of STATe<x> = 1 to 4

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:CALCULATION:STATE1 ON
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:CALCULATION:STATE1? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:CALCULATION:STATE1 1
```

:ANALysis:AHIStogram<x>:MEASure:CURSor:CALCulation:VALue<x>?

Function Queries the measured value of the calculation item of the cursor of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:CURSor:CALCulation:VALue<x>?
```
<x> of AHIStogram<x> = 1 or 2
<x> of VALue<x> = 1 to 4

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:CALCULATION:VALUE1? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:CALCULATION:VALUE1 1.000E+00
```

:ANALysis:AHIStogram<x>:MEASure:CURSor:HLINkage

Function Turns ON/OFF the horizontal cursor link of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:CURSor:HLINkage {<Boolean>}
:ANALysis:AHIStogram<x>:MEASure:CURSor:HLINkage?
```
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:HLINKAGE ON
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:HLINKAGE? -> :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:HLINKAGE 1
```

:ANALysis:AHIStogram<x>:MEASure:CURSor:HPOSition<x>

Function Sets the horizontal cursor position of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:CURSor:HPOSition<x> {<NRf>}
:ANALysis:AHIStogram<x>:MEASure:CURSor:HPOSition<x>?
```
<x> of AHIStogram<x> = 1 or 2
<x> of HPOSition<x> = 1 or 2
<NRf> = –5 to 5 div

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:HPOSITION1 1
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:HPOSITION1? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:HPOSITION1 1.000E+00
```

:ANALysis:AHIStogram<x>:MEASure:CURSor:VLINkage

Function Turns ON/OFF the vertical cursor link of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:CURSor:VLINkage {<Boolean>}
:ANALysis:AHIStogram<x>:MEASure:CURSor:VLINkage?
```
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:VLINKAGE ON
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:VLINKAGE? -> :ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:VLINKAGE 1
```

:ANALysis:AHIStogram<x>:MEASure:CURSor:VPOSition<x>

Function Sets the vertical cursor position of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:CURSor:VPOSition<x> {<NRf>}
:ANALysis:AHIStogram<x>:MEASure:CURSor:VPOSition<x>?
```
<x> of AHIStogram<x> = 1 and 2
<x> of VPOSition<x> = 1 and 2
<NRf> = –4 to 4 (div)

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:VPOSITION1 1
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:VPOSITION1? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:CURSOR:VPOSITION1 1.000E+00
```

:ANALysis:AHIStogram<x>:MEASure:MODE

Function Sets the automated measurement mode of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:MODE {CURSor|OFF|PARameter}
:ANALysis:AHIStogram<x>:MEASure:MODE?
```
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:MODE CURSOR
:ANALYSIS:AHISTOGRAM1:MEASURE:MODE? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:MODE CURSOR
```

:ANALysis:AHIStogram<x>:MEASure:PARameter?

Function Queries all settings related to the automated measurement of waveform parameters of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:PARameter?
```
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER? -> :ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA1:MAXIMUM:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA1:MEAN:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA1:MEDIAN:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA1:MINIMUM:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA1:PEAK:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA1:RMS:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA1:SD2INTEG:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA1:SD3INTEG:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA1:SDEVIATION:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA1:SDINTEG:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA2:MAXIMUM:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA2:MEAN:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA2:MEDIAN:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA2:MINIMUM:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA2:PEAK:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA2:RMS:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA2:SD2INTEG:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA2:SD3INTEG:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA2:SDEVIATION:STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:AREA2:SDINTEG:STATE 1.....
```

:ANALysis:AHIStogram<x>:MEASure:PARameter:AREA<x>?

Function Queries all settings related to the area of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
PARameter:AREA<x>?
```
<x> of AHIStogram<x> = 1 and 2
<x> of AREA<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:MAXIMUM:STATE 1;:
ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:
AREA1:MEAN:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:PARAMETER:AREA1:
MEDIAN:STATE 1;:ANALYSIS:AHISTOGRAM1:
MEASURE:PARAMETER:AREA1:MINIMUM:
STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:PEAK:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:PARAMETER:AREA1:
RMS:STATE 1;:ANALYSIS:AHISTOGRAM1:
MEASURE:PARAMETER:AREA1:SD2INTEG:
STATE 1;:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:SD3INTEG:STATE 1;:
ANALYSIS:AHISTOGRAM1:MEASURE:PARAMETER:
AREA1:SDEVIATION:STATE 1;:ANALYSIS:
AHISTOGRAM1:MEASURE:PARAMETER:AREA1:
SDINTEG:STATE 1
```

:ANALysis:AHIStogram<x>:MEASure:PARameter:AREA<x>:ALL

Function Turns ON/OFF all waveform parameters of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
PARameter:AREA<x>:ALL {<Boolean>}
```
<x> of AHIStogram<x> = 1 or 2
<x> of AREA<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:ALL ON
```

:ANALysis:AHIStogram<x>:MEASure:PARameter:AREA<x>:<Parameter>?

Function Queries all settings related to the waveform parameter of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
PARameter:AREA<x>:<Parameter>?
```
<x> of AHIStogram<x> = 1 or 2
<x> of AREA<x> = 1 or 2
<Parameter> = {MAXimum|MEAN|MEDian|MINimum|PEAK|RMS|SD2integ|SD3integ|SDEViation|SDINteg}

Example (The following is an example for the maximum value of area 1.)
```
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:MAXIMUM? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:MAXIMUM:STATE 1
```

:ANALysis:AHIStogram<x>:MEASure:PARameter:AREA<x>:<Parameter>:STATe

Function Turns ON/OFF the waveform parameter of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
PARameter:AREA<x>:<Parameter>:
STATe {<Boolean>}
:ANALysis:AHIStogram<x>:MEASure:
PARameter:AREA<x>:<Parameter>:STATe?
```
<x> of AHIStogram<x> = 1 or 2
<x> of AREA<x> = 1 or 2
<Parameter> = {MAXimum|MEAN|MEDian|MINimum|PEAK|RMS|SD2integ|SD3integ|SDEViation|SDINteg}

Example (The following is an example for the maximum value of area 1.)
```
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:MAXIMUM:STATE ON
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:MAXIMUM:STATE? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:MAXIMUM:STATE 1
```

:ANALysis:AHIStogram<x>:MEASure:PARameter:AREA<x>:<Parameter>:VALue?

Function Queries the automated measured value of the waveform parameter of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
PARameter:AREA<x>:<Parameter>:VALue?
```
<x> of AHIStogram<x> = 1 or 2
<x> of AREA<x> = 1 or 2
<Parameter> = {MAXimum|MEAN|MEDian|MINimum|PEAK|RMS|SD2integ|SD3integ|SDEViation|SDINteg}

Example (The following is an example for the maximum value of area 1.)
```
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:MAXIMUM:VALUE? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:AREA1:MAXIMUM:VALUE 1.000E+00
```

:ANALysis:AHIStogram<x>:MEASure:PARameter:CALCulation?

Function Queries all settings related to the calculation items of waveform parameters of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
PARameter:CALCulation?
```
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:CALCULATION? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:CALCULATION:DEFINE1 “MEAN”;
DEFINE2 “MAX”;DEFINE3 “MIN”;
DEFINE4 “PEAK”;STATE1 1;STATE2 1;
STATE3 1;STATE4 1
```

:ANALysis:AHIStogram<x>:MEASure:PARameter:CALCulation:ALL

Function Turns ON/OFF all calculation items of the waveform parameters of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
PARameter:CALCulation:ALL {<Boolean>}
```
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:CALCULATION:ALL ON
```

:ANALysis:AHIStogram<x>:MEASure:PARameter:CALCulation:DEFine<x>

Function Sets the equation of the calculation items of the waveform parameter of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
PARameter:CALCulation:
DEFine<x> {<String>}
:ANALysis:AHIStogram<x>:MEASure:
PARameter:CALCulation:DEFine<x>?
```
<x> of AHIStogram<x> = 1 or 2
<x> of DEFine<x> = 1 to 4
<String> = Up to 128 characters

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:CALCULATION:DEFINE1 “MEAN”
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:CALCULATION:DEFINE1? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:CALCULATION:DEFINE1 “MEAN”
```

:ANALysis:AHIStogram<x>:MEASure:PARameter:CALCulation:STATe<x>

Function Turns ON/OFF the calculation items of the waveform parameter of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
PARameter:CALCulation:STATe<x>
{<Boolean>}
:ANALysis:AHIStogram<x>:MEASure:
PARameter:CALCulation:STATe<x>?
```
<x> of AHIStogram<x> = 1 or 2
<x> of STATe<x> = 1 to 4

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:CALCULATION:STATE1 ON
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:CALCULATION:STATE1? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:CALCULATION:STATE1 1
```

:ANALysis:AHIStogram<x>:MEASure:PARameter:CALCulation:VALue<x>?

Function Queries the automated measured value of the calculation items of the waveform parameter of the accumulated histogram.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
PARameter:CALCulation:VALue<x>?
```
<x> of AHIStogram<x> = 1 or 2
<x> of VALue<x> = 1 to 4

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:CALCULATION:VALUE1? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:CALCULATION:VALUE1 1.000E+00
```

:ANALysis:AHIStogram<x>:MEASure:PARameter:HRANge<x>

Function Sets the horizontal range of the waveform parameter of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
PARameter:HRANge<x> {<NRf>,<NRf>}
:ANALysis:AHIStogram<x>:MEASure:
PARameter:HRANge<x>?
```
<x> of AHIStogram<x> = 1 or 2
<x> of HRANge<x> = 1 or 2
<NRf> = –5 to 5 div

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:HRANGE1 1,2
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:HRANGE1? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:HRANGE1 2.000E+00,1.000E+00
```

:ANALysis:AHIStogram<x>:MEASure:PARameter:VRANge<x>

Function Sets the vertical range of the waveform parameter of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MEASure:
PARameter:VRANge<x> {<NRf>,<NRf>}
:ANALysis:AHIStogram<x>:MEASure:
PARameter:VRANge<x>?
```
<x> of AHIStogram<x> = 1 or 2
<x> of VRANge<x> = 1 or 2
<NRf> = –4 to 4 (div)

Example
```
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:VRANGE1 1,2
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:VRANGE1? ->
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:VRANGE1 2.000E+00,1.000E+00
```

:ANALysis:AHIStogram<x>:MODE

Function Sets the accumulated histogram mode or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:MODE
{HORizontal|VERTical}
:ANALysis:AHIStogram<x>:MODE?
```
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:MODE HORIZONTAL
:ANALYSIS:AHISTOGRAM1:MODE? ->
:ANALYSIS:AHISTOGRAM1:MODE HORIZONTAL
```

:ANALysis:AHIStogram<x>:TRACe

Function Sets the source trace of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:TRACe {<NRf>}
:ANALysis:AHIStogram<x>:TRACe?
```
<x> = 1 or 2
<NRf> = 1 to 8

Example
```
:ANALYSIS:AHISTOGRAM1:TRACE 1
:ANALYSIS:AHISTOGRAM1:TRACE? ->
:ANALYSIS:AHISTOGRAM1:TRACE 1
```

:ANALysis:AHIStogram<x>:VERTical

Function Sets the vertical range of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:VERTical
{<NRf>,<NRf>}
:ANALysis:AHIStogram<x>:VERTical?
```
<x> = 1 or 2
<NRf> = –4 to 4 (div)

Example
```
:ANALYSIS:AHISTOGRAM1:VERTICAL 1,2
:ANALYSIS:AHISTOGRAM1:VERTICAL? ->
:ANALYSIS:AHISTOGRAM1:
VERTICAL 2.000E+00,1.000E+00
```

:ANALysis:AHIStogram<x>:WINDow

Function Sets the measurement target window of the accumulated histogram or queries the current setting.

Syntax
```
:ANALysis:AHIStogram<x>:WINDow {MAIN|
Z1|Z2}
:ANALysis:AHIStogram<x>:WINDow?
```
<x> = 1 or 2

Example
```
:ANALYSIS:AHISTOGRAM1:WINDOW MAIN
:ANALYSIS:AHISTOGRAM1:WINDOW? ->
:ANALYSIS:AHISTOGRAM1:WINDOW MAIN
```

:ANALysis:DISPlay<x>

Function Turns ON/OFF the analysis function display or queries the current setting.

Syntax
```
:ANALysis:DISPlay<x> {<Boolean>}
:ANALysis:DISPlay<x>?
```
<x> = 1 or 2

Example
```
:ANALYSIS:DISPLAY1 ON
:ANALYSIS:DISPLAY1? ->
:ANALYSIS:DISPLAY1 1
```

:ANALysis:FFT<x>?

Function Queries all settings related to the FFT computation function.

Syntax
```
:ANALysis:FFT<x>?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1? ->
:ANALYSIS:FFT1:HORIZONTAL:CSPAN:
CENTER 1.000E+00;SPAN1.000E+00;
:ANALYSIS:FFT1:HORIZONTAL:
LRIGHT:RANGE2.000E+00,1.000E+00;
:ANALYSIS:FFT1:HORIZONTAL:MODE AUTO;
:ANALYSIS:FFT1:IPART 1;LENGTH 2500;
MAXHOLD 1;MEASURE:MARKer:BASIC:
DFREQUENCY:STATE1;:ANALYSIS:FFT1:
MEASURE:MARKER:BASIC:DV:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY1:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY2:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
POSITION1 1.000E+00;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
POSITION2 2.000E+00;V1:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
V2:STATE 1;:ANALYSIS:FFT1:MEASURE:
MARKER:CALCULATION:DEFINE1 "V(F1)";
DEFINE2 "V(F2)";DEFINE3 "V(F1)";
DEFINE4 "V(F2)";STATE1 1;STATE2 1;
STATE3 1;STATE4 1;
:ANALYSIS:FFT1:MEASURE:MODE MARKER;
PEAK:BASIC:DFREQUENCY:STATE 1;
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
DV:STATE 1;
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
FREQUENCY1:STATE 1;
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
FREQUENCY2:STATE 1;
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
RANGE1 1.000E+00,0.000E+00;
RANGE2 1.000E+00,0.000E+00;V1:
STATE 1;
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:V2:
STATE 1.....
```

:ANALysis:FFT<x>:HORizontal?

Function Queries all settings related the horizontal axis of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:HORizontal?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:HORIZONTAL? ->
:ANALYSIS:FFT1:HORIZONTAL:CSPAN:
CENTER 1.000E+00;SPAN 1.000E+00;:
ANALYSIS:FFT1:HORIZONTAL:LRIGHT:
RANGE 2.000E+00,1.000E+00;:ANALYSIS:
FFT1:HORIZONTAL:MODE AUTO
```

:ANALysis:FFT<x>:HORizontal:CSPan?

Function Queries all settings related to the center and span of the horizontal axis of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:HORizontal:CSPan?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:HORIZONTAL:CSPAN? ->
:ANALYSIS:FFT1:HORIZONTAL:CSPAN:
CENTER 1.000E+00;SPAN 1.000E+00
```

:ANALysis:FFT<x>:HORizontal:CSPan:CENTer

Function Sets the horizontal center of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:HORizontal:CSPan:
CENTer {<Frequency>}
:ANALysis:FFT<x>:HORizontal:CSPan:
CENTer?
```
<x> = 1 or 2
<Frequency> = 0 to 250 G (Hz)

Example
```
:ANALYSIS:FFT1:HORIZONTAL:CSPAN:
CENTER 1HZ
:ANALYSIS:FFT1:HORIZONTAL:CSPAN:CENTER?
-> :ANALYSIS:FFT1:HORIZONTAL:CSPAN:
CENTER 1.000E+00
```

:ANALysis:FFT<x>:HORizontal:CSPan:SPAN

Function Sets the horizontal span of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:HORizontal:CSPan:
SPAN {<Frequency>}
:ANALysis:FFT<x>:HORizontal:CSPan:SPAN?
```
<x> = 1 or 2
<Frequency> = 0 to 250 G (Hz)

Example
```
:ANALYSIS:FFT1:HORIZONTAL:CSPAN:
SPAN 1HZ
:ANALYSIS:FFT1:HORIZONTAL:CSPAN:SPAN?
-> :ANALYSIS:FFT1:HORIZONTAL:CSPAN:
SPAN 1.000E+00
```

:ANALysis:FFT<x>:HORizontal:LRIGht?

Function Queries all settings related the left and right edges of the horizontal axis of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:HORizontal:LRIGht?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:HORIZONTAL:LRIGHT? ->
:ANALYSIS:FFT1:HORIZONTAL:LRIGHT:
RANGE 2.000E+00,1.000E+00
```

:ANALysis:FFT<x>:HORizontal:LRIGht:RANGe

Function Sets the range of the horizontal left and right edges of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:HORizontal:LRIGht:
RANGe {<Frequency>,<Frequency>}
:ANALysis:FFT<x>:HORizontal:LRIGht:
RANGe?
```
<x> = 1 or 2
<Frequency> = 0 to 250 G (Hz)

Example
```
:ANALYSIS:FFT1:HORIZONTAL:LRIGHT:
RANGE 1HZ,2HZ
:ANALYSIS:FFT1:HORIZONTAL:LRIGHT:RANGE?
-> :ANALYSIS:FFT1:HORIZONTAL:LRIGHT:
RANGE 2.000E+00,1.000E+00
```

:ANALysis:FFT<x>:HORizontal:MODE

Function Sets the horizontal mode of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:HORizontal:MODE {AUTO|
CSPan|LRIGht}
:ANALysis:FFT<x>:HORizontal:MODE?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:HORIZONTAL:MODE AUTO
:ANALYSIS:FFT1:HORIZONTAL:MODE? ->
:ANALYSIS:FFT1:HORIZONTAL:MODE AUTO
```

:ANALysis:FFT<x>:IPARt (Imag Part)

Function Sets the source trace of the imaginary part of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:IPARt {<NRf>|DONTcare}
:ANALysis:FFT<x>:IPARt?
```
<x> = 1 or 2
<NRf> = 1 to 8

Example
```
:ANALYSIS:FFT1:IPART 1
:ANALYSIS:FFT1:IPART? ->
:ANALYSIS:FFT1:IPART 1
```

:ANALysis:FFT<x>:LENGth

Function Sets the number of FFT points of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:LENGth {<NRf>}
:ANALysis:FFT<x>:LENGth?
```
<x> = 1 or 2
<NRf> = 2500, 6250, 12500, 25000, 62500, 125000, or 250000

Example
```
:ANALYSIS:FFT1:LENGTH 2500
:ANALYSIS:FFT1:LENGTH? ->
:ANALYSIS:FFT1:LENGTH 2500
```

:ANALysis:FFT<x>:MAXHold

Function Turns ON/OFF the maximum value hold function of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MAXHold {<Boolean>}
:ANALysis:FFT<x>:MAXHold?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MAXHOLD ON
:ANALYSIS:FFT1:MAXHOLD? ->
:ANALYSIS:FFT1:MAXHOLD 1
```

:ANALysis:FFT<x>:MEASure?

Function Queries all settings related to the automated measurement of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE? ->
:ANALYSIS:FFT1:MEASURE:MARKer:BASIC:
DFREQUENCY:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
DV:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY1:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY2:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
POSITION1 1.000E+00;
:ANALYSIS:FFT1:MEASURE:
MARKER:BASIC:POSITION2
2.000E+00;V1:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
V2:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:
CALCULATION:DEFINE1 "V(F1)";DEFINE2
"V(F2)";DEFINE3 "V(F1)";DEFINE4
"V(F2)";STATE1 1;STATE2 1;STATE3
1;STATE4 1;
:ANALYSIS:FFT1:MEASURE:MODE
MARKER;PEAK:BASIC:DFREQUENCY:STATE 1;
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
DV:STATE 1;
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
FREQUENCY1:STATE 1;
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
FREQUENCY2:STATE 1;
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
RANGE1 1.000E+00,0.000E+00;RANGE2
1.000E+00,0.000E+00;V1:STATE 1;
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
V2:STATE 1;
:ANALYSIS:FFT1:MEASURE:PEAK:
CALCULATION:DEFINE1 "V(P1)";DEFINE2
"V(P2)";DEFINE3 "V(P1)";DEFINE4
"V(P2)";STATE1 1;STATE2 1;STATE3
1;STATE4 1
```

:ANALysis:FFT<x>:MEASure:MARKer?

Function Queries all settings related to the marker cursor measurement of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:MARKer?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER? ->
:ANALYSIS:FFT1:MEASURE:MARKer:BASIC:
DFREQUENCY:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
DV:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY1:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY2:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
POSITION1 1.000E+00;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
POSITION2 2.000E+00;V1:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
V2:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:
CALCULATION:DEFINE1 "V(F1)";DEFINE2
"V(F2)";DEFINE3 "V(F1)";DEFINE4
"V(F2)";STATE1 1;STATE2 1;STATE3
1;STATE4 1
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]?

Function Queries all settings related to basic items of the marker cursor of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC? ->
:ANALYSIS:FFT1:MEASURE:MARKer:BASIC:
DFREQUENCY:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:
BASIC:DV:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY1:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY2:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:
BASIC:POSITION1 1.000E+00;
:ANALYSIS:FFT1:MEASURE:
MARKER:BASIC:POSITION2
2.000E+00;V1:STATE 1;
:ANALYSIS:FFT1:MEASURE:MARKER:
BASIC:V2:STATE 1
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]:ALL

Function Turns ON/OFF all basic items of the marker cursor of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:ALL {<Boolean>}
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
ALL ON
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]:DFRequency?

Function Queries all settings related to the frequency value between marker cursors of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:DFRequency?
```
<x> = 1, 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKer:BASIC:
DFREQUENCY? ->
:ANALYSIS:FFT1:MEASURE:MARKer:BASIC:
DFREQUENCY:STATE 1
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]:DFRequency:STATe

Function Turns ON/OFF the frequency value between marker cursors of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:DFRequency:
STATe {<Boolean>}
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:DFRequency:STATe?
```
<x> = 1, 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKer:BASIC:
DFREQUENCY:STATE ON
:ANALYSIS:FFT1:MEASURE:MARKer:BASIC:
DFREQUENCY:STATE? ->
:ANALYSIS:FFT1:MEASURE:MARKer:BASIC:
DFREQUENCY:STATE 1
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]:DFRequency:VALue?

Function Queries the frequency value between marker cursors of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:DFRequency:VALue?
```
<x> = 1, 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKer:BASIC:
DFREQUENCY:VALUE? ->
:ANALYSIS:FFT1:MEASURE:MARKer:BASIC:
DFREQUENCY:VALUE 1.000E+00
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]:DV?

Function Queries all settings related to the power value between marker cursors of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:DV?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:DV?
-> :ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
DV:STATE 1
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]:DV:STATe

Function Turns ON/OFF the power value between marker cursors of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:DV:STATe {<Boolean>}
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:DV:STATe?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:DV:
STATE ON
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:DV:
STATE? -> :ANALYSIS:FFT1:MEASURE:
MARKER:BASIC:DV:STATE 1
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]:DV:VALue?

Function Queries the power value between marker cursors of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:DV:VALue?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:DV:
VALUE? -> :ANALYSIS:FFT1:MEASURE:
MARKER:BASIC:DV:VALUE 1.000E+00
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]:FREQuency<x>?

Function Queries all settings related to the frequency value of the marker cursor of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:FREQuency<x>?
```
<x> of FFT<x> = 1, 2
<x> of FREQuency<x> = 1, 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY1? ->
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY1:STATE 1
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]:FREQuency<x>:STATe

Function Turns ON/OFF the frequency value of the marker cursor of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:FREQuency<x>:
STATe {<Boolean>}
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:FREQuency<x>:STATe?
```
<x> of FFT<x> = 1, 2
<x> of FREQuency<x> = 1, 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY1:STATE ON
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY1:STATE? ->
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY1:STATE 1
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]:FREQuency<x>:VALue?

Function Queries the frequency value of the marker cursor of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:FREQuency<x>:VALue?
```
<x> of FFT<x> = 1, 2
<x> of FREQuency<x> = 1, 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY1:VALUE? ->
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
FREQUENCY1:VALUE 1.000E+00
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]:POSition<x> {<NRf>}

Function Sets the marker cursor position of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:POSition<x> {<NRf>}
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:POSition<x>?
```
<x> of FFT<x> = 1, 2
<x> of POSition<x> = 1, 2
<NRf> = –5 to 5 (div)

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
POSITION1 1
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
POSITION1? ->
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:
POSITION1 1.000E+00
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]:V<x>?

Function Queries all settings related to the power value of the marker cursor of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:V<x>?
```
<x> of FFT<x> = 1 or 2
<x> of V<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:V1?
-> :ANALYSIS:FFT1:MEASURE:MARKER:
BASIC:STATE 1
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]:V<x>:STATe

Function Turns ON/OFF the power value of the marker cursor of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:V<x>:STATe {<Boolean>}
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:V<x>:STATe?
```
<x> of FFT<x> = 1 or 2
<x> of V<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:V1:
STATE ON
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:V1:
STATE? -> :ANALYSIS:FFT1:MEASURE:
MARKER:BASIC:V1:STATE 1
```

:ANALysis:FFT<x>:MEASure: MARKer[:BASic]:V<x>:VALue?

Function Queries the power value of the marker cursor of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:
MARKer[:BASic]:V<x>:VALue?
```
<x> of FFT<x> = 1 or 2
<x> of V<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:BASIC:V1:
VALUE? -> :ANALYSIS:FFT1:MEASURE:
MARKER:BASIC:V1:VALUE 1.000E+00
```

:ANALysis:FFT<x>:MEASure:MARKer: CALCulation?

Function Queries all settings related to calculation items of the marker cursor of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:MARKer:
CALCulation?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:
CALCULATION? ->
:ANALYSIS:FFT1:MEASURE:MARKER:
CALCULATION:DEFINE1 “V(F1)”;
DEFINE2 “V(F2)”;DEFINE3 “V(F1)”;D
EFINE4 “V(F2)”;STATE1 1;STATE2 1;
STATE3 1;STATE4 1
```

:ANALysis:FFT<x>:MEASure:MARKer: CALCulation:ALL

Function Turns ON/OFF all calculation items of the marker cursor of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:MARKer:
CALCulation:ALL {<Boolean>}
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:
CALCULATION:ALL ON
```

:ANALysis:FFT<x>:MEASure:MARKer: CALCulation:DEFine<x>

Function Sets the equation of the calculation items of the marker cursor of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:MARKer:
CALCulation:DEFine<x> {<String>}
:ANALysis:FFT<x>:MEASure:MARKer:
CALCulation:DEFine<x>?
```
<x> of FFT<x> = 1 or 2
<x> of DEFine<x> = 1 to 4
<String> = Up to 128 characters

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:
CALCULATION:DEFINE1 “V(F1)”
:ANALYSIS:FFT1:MEASURE:MARKER:
CALCULATION:DEFINE1? ->
:ANALYSIS:FFT1:MEASURE:MARKER:
CALCULATION:DEFINE1 “V(F1)”
```

:ANALysis:FFT<x>:MEASure:MARKer: CALCulation:STATe<x>

Function Turns ON/OFF the calculation items of the marker cursor of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:MARKer:C
ALCulation:STATe<x> {<Boolean>}
:ANALysis:FFT<x>:MEASure:MARKer:
CALCulation:STATe<x>?
```
<x> of FFT<x> = 1 or 2
<x> of STATe<x> = 1 to 4

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:
CALCULATION:STATE1 ON
:ANALYSIS:FFT1:MEASURE:MARKER:
CALCULATION:STATE1? ->
:ANALYSIS:FFT1:MEASURE:MARKER:
CALCULATION:STATE1 1
```

:ANALysis:FFT<x>:MEASure:MARKer: CALCulation:VALue<x>?

Function Queries the measured value of the calculation items of the marker cursor of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:MARKer:
CALCulation:VALue<x>?
```
<x> of FFT<x> = 1 or 2
<x> of VALue<x> = 1 to 4

Example
```
:ANALYSIS:FFT1:MEASURE:MARKER:
CALCULATION:VALUE1? ->
:ANALYSIS:FFT1:MEASURE:MARKER:
CALCULATION:VALUE1 1.000E+00
```

:ANALysis:FFT<x>:MEASure:MODE

Function Sets the automated measurement mode of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:MODE {MARKer|
OFF|PEAK}
:ANALysis:FFT<x>:MEASure:MODE?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:MODE MARKER
:ANALYSIS:FFT1:MEASURE:MODE? ->
:ANALYSIS:FFT1:MEASURE:MODE MARKER
```

:ANALysis:FFT<x>:MEASure:PEAK?

Function Queries all settings related to the peak value measurement of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK? ->
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
DFREQUENCY:STATE 1;:ANALYSIS:FFT1:
MEASURE:PEAK:BASIC:DV:STATE 1;:
ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
FREQUENCY1:STATE 1;:ANALYSIS:FFT1:
MEASURE:PEAK:BASIC:FREQUENCY2:STATE 1;:
ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
RANGE1 1.000E+00,0.000E+00;
RANGE2 1.000E+00,0.000E+00;V1:STATE 1;:
ANALYSIS:FFT1:MEASURE:PEAK:BASIC:V2:
STATE 1;:ANALYSIS:FFT1:MEASURE:PEAK:
CALCULATION:DEFINE1 “V(P1)”;
DEFINE2 “V(P2)”;DEFINE3 “V(P1)”;
DEFINE4 “V(P2)”;STATE1 1;STATE2 1;
STATE3 1;STATE4 1
```

:ANALysis:FFT<x>:MEASure: PEAK[:BASic]?

Function Queries all settings related to basic items of the peak value of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC? ->
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
DFREQUENCY:STATE 1;:ANALYSIS:FFT1:
MEASURE:PEAK:BASIC:DV:STATE 1;:
ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
FREQUENCY1:STATE 1;:ANALYSIS:FFT1:
MEASURE:PEAK:BASIC:FREQUENCY2:STATE
1;:
ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
RANGE1 1.000E+00,0.000E+00;
RANGE2 1.000E+00,0.000E+00;V1:
STATE 1;:ANALYSIS:FFT1:MEASURE:PEAK:
BASIC:V2:STATE 1
```

:ANALysis:FFT<x>:MEASure: PEAK[:BASic]:ALL

Function Turns ON/OFF all basic items of the peak value of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:
PEAK[:BASic]:ALL {<Boolean>}
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
ALL ON
```

:ANALysis:FFT<x>:MEASure: PEAK[:BASic]:DFRequency?

Function Queries all settings related to the frequency value between peak values of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:
DFRequency?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
DFREQUENCY? ->
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
DFREQUENCY:STATE 1
```

:ANALysis:FFT<x>:MEASure: PEAK[:BASic]:DFRequency:STATe

Function Turns ON/OFF the frequency value between peak values of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:
DFRequency:STATe {<Boolean>}
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:
DFRequency:STATe?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
DFREQUENCY:STATE ON
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
DFREQUENCY:STATE? ->
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
DFREQUENCY:STATE 1
```

:ANALysis:FFT<x>:MEASure: PEAK[:BASic]:DFRequency:VALue?

Function Queries the frequency value between peak values of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:
DFRequency:VALue?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
DFREQUENCY:VALUE? ->
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
DFREQUENCY:VALUE 1.000E+00
```

:ANALysis:FFT<x>:MEASure: PEAK[:BASic]:DV?

Function Queries all settings related to the power value between peak values of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:
DV?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:DV?
-> :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
DV:STATE 1
```

:ANALysis:FFT<x>:MEASure: PEAK[:BASic]:DV:STATe

Function Turns ON/OFF the power value between peak values of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:
DV:STATe {<Boolean>}
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:
DV:STATe?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:DV:
STATE ON
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:DV:
STATE? -> :ANALYSIS:FFT1:MEASURE:PEAK:
BASIC:DV:STATE 1
```

:ANALysis:FFT<x>:MEASure: PEAK[:BASic]:DV:VALue?

Function Queries the power value between peak values of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:
DV:VALue?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:DV:
VALUE? -> :ANALYSIS:FFT1:MEASURE:PEAK:
BASIC:DV:VALUE 1.000E+00
```

:ANALysis:FFT<x>:MEASure: PEAK[:BASic]:FREQuency<x>?

Function Queries all settings related to the peak frequency value of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:
FREQuency<x>?
```
<x> of FFT<x> = 1 or 2
<x> of FREQuency<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
FREQUENCY1? -> :ANALYSIS:FFT1:MEASURE:
PEAK:BASIC:FREQUENCY1:STATE 1
```

:ANALysis:FFT<x>:MEASure: PEAK[:BASic]:FREQuency<x>:STATe

Function Turns ON/OFF the peak frequency value of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:F
REQuency<x>:STATe {<Boolean>}
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:
FREQuency<x>:STATe?
```
<x> of FFT<x> = 1 or 2
<x> of FREQuency<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
FREQUENCY1:STATE ON
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
FREQUENCY1:STATE? -> :ANALYSIS:FFT1:
MEASURE:PEAK:BASIC:FREQUENCY1:STATE 1
```

:ANALysis:FFT<x>:MEASure: PEAK[:BASic]:FREQuency<x>:VALue?

Function Queries the peak frequency value of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:
FREQuency<x>:VALue?
```
<x> of FFT<x> = 1 or 2
<x> of FREQuency<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
FREQUENCY1:VALUE? ->
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:
FREQUENCY1:VALUE 1.000E+00
```

:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:RANGe<x>

Function Sets the measurement range of the peak value of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:RANGe<x> {<NRf>,<NRf>}
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:RANGe<x>?
```
<x> of FFT<x> = 1 or 2
<x> of RANGe<x> = 1 or 2
<NRf> = –5 to 5 div

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:RANGE1 0,1
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:RANGE1? -> :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:RANGE1 1.000E+00,0.000E+00
```

:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:V<x>?

Function Queries all settings related to the peak value of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:V<x>?
```
<x> of FFT<x> = 1 or 2
<x> of V<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:V1? -> :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:V1:STATE 1
```

:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:V<x>:STATe

Function Turns ON/OFF the peak value of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:V<x>:STATe {<Boolean>}
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:V<x>:STATe?
```
<x> of FFT<x> = 1 or 2
<x> of V<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:V1:STATE ON
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:V1:STATE? -> :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:V1:STATE 1
```

:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:V<x>:VALue?

Function Queries the peak value of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK[:BASic]:V<x>:VALue?
```
<x> of FFT<x> = 1 or 2
<x> of V<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:BASIC:V1:VALUE? -> :ANALYSIS:FFT1:MEASURE:PEAK:BASIC:V1:VALUE 1.000E+00
```

:ANALysis:FFT<x>:MEASure:PEAK:CALCulation?

Function Queries all settings related to calculation items of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK:CALCulation?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:CALCULATION? -> :ANALYSIS:FFT1:MEASURE:PEAK:CALCULATION:DEFINE1 “V(P1)”;DEFINE2 “V(P2)”;DEFINE3 “V(P1)”;DEFINE4 “V(P2)”;STATE1 1;STATE2 1;STATE3 1;STATE4 1
```

:ANALysis:FFT<x>:MEASure:PEAK:CALCulation:ALL

Function Turns ON/OFF all calculation items of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK:CALCulation:ALL {<Boolean>}
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:CALCULATION:ALL ON
```

:ANALysis:FFT<x>:MEASure:PEAK:CALCulation:DEFine<x>

Function Sets the equation of the calculation item of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK:CALCulation:DEFine<x> {<String>}
:ANALysis:FFT<x>:MEASure:PEAK:CALCulation:DEFine<x>?
```
<x> of FFT<x> = 1 or 2
<x> of DEFine<x> = 1 to 4
<String> = Up to 128 characters

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:CALCULATION:DEFINE1 “V(P1)”
:ANALYSIS:FFT1:MEASURE:PEAK:CALCULATION:DEFINE1? -> :ANALYSIS:FFT1:MEASURE:PEAK:CALCULATION:DEFINE1 “V(P1)”
```

:ANALysis:FFT<x>:MEASure:PEAK: CALCulation:STATe<x>

Function Turns ON/OFF the calculation items of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK:
CALCulation:STATe<x> {<Boolean>}
:ANALysis:FFT<x>:MEASure:PEAK:
CALCulation:STATe<x>?
```
<x> of FFT<x> = 1 or 2
<x> of STATe<x> = 1 to 4

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:
CALCULATION:STATE1 ON
:ANALYSIS:FFT1:MEASURE:PEAK:
CALCULATION:STATE1? ->
:ANALYSIS:FFT1:MEASURE:PEAK:
CALCULATION:STATE1 1
```

:ANALysis:FFT<x>:MEASure:PEAK: CALCulation:VALue<x>?

Function Queries the measured value of the calculation item of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:MEASure:PEAK:
CALCulation:VALue<x>?
```
<x> of FFT<x> = 1 or 2
<x> of VALue<x> = 1 to 4

Example
```
:ANALYSIS:FFT1:MEASURE:PEAK:
CALCULATION:VALUE1? ->
:ANALYSIS:FFT1:MEASURE:PEAK:
CALCULATION:VALUE1 1.000E+00
```

:ANALysis:FFT<x>:RANGe

Function Sets the measurement source window used in the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:RANGe {MAIN|Z1|Z2}
:ANALysis:FFT<x>:RANGe?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:RANGE MAIN
:ANALYSIS:FFT1:RANGE? ->
:ANALYSIS:FFT1:RANGE MAIN
```

:ANALysis:FFT<x>:RPARt (Real Part)

Function Sets the source trace of the real part of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:RPARt {<NRf>}
:ANALysis:FFT<x>:RPARt?
```
<x> = 1 or 2
<NRf> = 1 to 8

Example
```
:ANALYSIS:FFT1:RPART 1
:ANALYSIS:FFT1:RPART? ->
:ANALYSIS:FFT1:RPART 1
```

:ANALysis:FFT<x>:RPOSition (Ref Position)

Function Sets the center point of magnification of the vertical axis of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:RPOSition {<NRf>}
:ANALysis:FFT<x>:RPOSition?
```
<x> = 1 or 2
<NRf> = –4 to 4 (div)

Example
```
:ANALYSIS:FFT1:RPOSITION 1
:ANALYSIS:FFT1:RPOSITION? ->
:ANALYSIS:FFT1:RPOSITION 1.000E+00
```

:ANALysis:FFT<x>:VERTical?

Function Queries all settings related the vertical axis of the FFT computation.

Syntax
```
:ANALysis:FFT<x>:VERTical?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:VERTICAL? ->
:ANALYSIS:FFT1:VERTICAL:
LEVEL 1.000E+00;MODE AUTO;
SENSITIVITY 1.000E+00
```

:ANALysis:FFT<x>:VERTical:LEVel

Function Sets the display position of the vertical axis of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:VERTical:LEVel {<NRf>}
:ANALysis:FFT<x>:VERTical:LEVel?
```
<x> = 1 or 2
<NRf> = –1.000E+31 to 1.000E+31 (dBV)

Example
```
:ANALYSIS:FFT1:VERTICAL:LEVEL 1
:ANALYSIS:FFT1:VERTICAL:LEVEL? ->
:ANALYSIS:FFT1:VERTICAL:LEVEL 1.000E+00
```

:ANALysis:FFT<x>:VERTical:MODE

Function Sets the vertical axis mode of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:VERTical:MODE {AUTO|
MANual}
:ANALysis:FFT<x>:VERTical:MODE?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:VERTICAL:MODE AUTO
:ANALYSIS:FFT1:VERTICAL:MODE? ->
:ANALYSIS:FFT1:VERTICAL:MODE AUTO
```

:ANALysis:FFT<x>:VERTical:SENSitivity

Function Sets the vertical sensitivity of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:VERTical:SENSitivity {<NRf>}
:ANALysis:FFT<x>:VERTical:SENSitivity?
```
<x> = 1 or 2
<NRf> = 0 to 1.000E+31 (dBV)

Example
```
:ANALYSIS:FFT1:VERTICAL:SENSITIVITY 1
:ANALYSIS:FFT1:VERTICAL:SENSITIVITY? ->
:ANALYSIS:FFT1:VERTICAL:SENSITIVITY
1.000E+00
```

:ANALysis:FFT<x>:WINDow

Function Sets the window function of the FFT computation or queries the current setting.

Syntax
```
:ANALysis:FFT<x>:WINDow {FLATtop|HANNing|RECTangle}
:ANALysis:FFT<x>:WINDow?
```
<x> = 1 or 2

Example
```
:ANALYSIS:FFT1:WINDOW FLATTOP
:ANALYSIS:FFT1:WINDOW? ->
:ANALYSIS:FFT1:WINDOW FLATTOP
```

:ANALysis:TYPE<x>

Function Sets the analysis function type or queries the current setting.

Syntax
```
:ANALysis:TYPE<x> {AHIStogram|FFT|WPARameter|XY}
:ANALysis:TYPE<x>?
```
<x> = 1 or 2

Example
```
:ANALYSIS:TYPE1 AHISTOGRAM
:ANALYSIS:TYPE1? ->
:ANALYSIS:TYPE1 AHISTOGRAM
```

:ANALysis:VTDisplay

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

Syntax
```
:ANALysis:VTDisplay {<Boolean>}
:ANALysis:VTDisplay?
```

Example
```
:ANALYSIS:VTDISPLAY ON
:ANALYSIS:VTDISPLAY? ->
:ANALYSIS:VTDISPLAY 1
```

:ANALysis:WAIT<x>?

Function Waits for the completion of the automated measurement with a timeout option.

Syntax
```
ANALysis:WAIT<x>? {<NRf>}
```
<x> = 1 or 2
<NRf> = 1 to 360000 (timeout value, in units of 10 ms)

Example
```
ANALYSIS:WAIT1? 100 ->:ANALYSIS:WAIT1 1
```

Description If the execution of the automated measurement completes within the timeout value, 0 is returned; if it is not complete or automated measurement is not being executed, 1 is returned. Even if the timeout value is set long, 0 is returned when the execution of the automated measurement is complete.

:ANALysis:WPARameter<x>?

Function Queries all settings related to the waveform parameter measurement function.

Syntax
```
:ANALysis:WPARameter<x>?
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1? ->
:ANALYSIS:WPARAMETER1:CALCULATION 1;
HISTOGRAM:MEASURE:MODE CURSOR;
CURSOR:C1:POSITION 1.000E+00;STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C2:
POSITION 2.000E+00;STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:DC:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:LINKAGE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:MEAN:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:PEAK:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:SD3INTEG:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:SDEVIATION:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:SDINTEG:STATE 1;
:ANALYSIS:WPARAMETER1:LIST:SCROLL
HORIZONTAL;:ANALYSIS:WPARAMETER1:
MODE HISTOGRAM;TREND:HSPAN 1;
MEASURE:CURSOR:C1:POSITION 1.000E+00;
STATE 1;
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
CURSOR:C2:POSITION 2.000E+00;STATE 1;
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
CURSOR:DC:STATE 1;
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
MODE CURSOR;
:ANALYSIS:WPARAMETER1:TREND:
VERTICAL 2.000E+00,1.000E+00
```

:ANALysis:WPARameter<x>:CALCulation

Function Sets the calculation items of the automated measurement of waveform parameters or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:CALCulation
{<NRf>}
:ANALysis:WPARameter<x>:CALCulation?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:ANALYSIS:WPARAMETER1:CALCULATION 1
:ANALYSIS:WPARAMETER1:CALCULATION? ->
:ANALYSIS:WPARAMETER1:CALCULATION 1
```

:ANALysis:WPARameter<x>:HISTogram?

Function Queries all settings related to the histogram display of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram?
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:MODE CURSOR;CURSOR:C1:
POSITION 1.000E+00;STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C2:
POSITION 2.000E+00;STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:DC:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:LINKAGE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:MEAN:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:PEAK:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:SD3INTEG:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:SDEVIATION:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:SDINTEG:STATE 1
```

:ANALysis:WPARameter<x>:HISTogram:MEASure?

Function Queries all settings related to the automated measurement of the histogram display of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure?
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:MODE CURSOR;CURSOR:C1:
POSITION 1.000E+00;STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C2:POSITION 2.000E+00;
STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:DC:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:LINKAGE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:MEAN:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:PEAK:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:SD3INTEG:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:SDEVIATION:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:SDINTEG:STATE 1
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor?

Function Queries all settings related to the cursor measurement on the histogram of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure:CURSor?
```
<x> = 1, 2

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C1:POSITION 1.000E+00;
STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C2:POSITION 2.000E+00;
STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:DC:STATE 1;
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:LINKAGE 1
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:ALL

Function Turns ON/OFF all histogram cursors of waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>HISTogram:
MEASure:CURSor:ALL {<Boolean>}
```
<x> = 1, 2

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:ALL ON
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:C<x>?

Function Queries all settings related to the cursor measurement on the histogram of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure:CURSor:C<x>?
```
<x> of WPARameter<x> = 1, 2
<x> of C<x> = 1, 2

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C1? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C1:POSITION 1.000E+00;
STATE 1
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:C<x>:POSition

Function Sets the cursor position on the trend of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure:PARameter:C<x>:POSition {<NRf>}
:ANALysis:WPARameter<x>:HISTogram:
MEASure:PARameter:C<x>:POSition?
```
<x> of WPARameter<x> = 1, 2
<x> of C<x> = 1, 2
<NRf> = –5 to 5 (div)

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C1:POSITION ON
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C1:POSITION? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C1:POSITION 1.000E+00
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:C<x>:STATe

Function Turns ON/OFF the cursor on the histogram of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure:PARameter:C<x>:
STATe {<Boolean>}
:ANALysis:WPARameter<x>:HISTogram:
MEASure:PARameter:C<x>:STATe?
```
<x> of WPARameter<x> = 1, 2
<x> of C<x> = 1, 2

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C1:STATE ON
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C1:STATE? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C1:STATE 1
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:C<x>:VALue?

Function Queries the measured value of the cursor on the histogram of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure:CURSor:C<x>:VALue?
```
<x> of WPARameter<x> = 1, 2
<x> of C<x> = 1, 2

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C1:VALUE? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:C1:VALUE 1.000E+00
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:DC?

Function Queries all settings related to the measurement between cursors on the histogram of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure:CURSor:DC?
```
<x> = 1, 2

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:DC? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:DC:STATE 1
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:DC:STATe

Function Turns ON/OFF the measurement between cursors on the histogram of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure:PARameter:DC:STATe {<Boolean>}
:ANALysis:WPARameter<x>:HISTogram:
MEASure:PARameter:DC:STATe?
```
<x> = 1, 2

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:DC:STATE ON
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:DC:STATE? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:DC:STATE 1
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:DC:VALue?

Function Queries the measured value between cursors on the histogram of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure:CURSor:DC:VALue?
```
<x> = 1, 2

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:DC:VALUE? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:DC:VALUE 1.000E+00
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:CURSor:LINKage

Function Turns ON/OFF the histogram cursor link of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure:CURSor:LINKage {<Boolean>}
:ANALysis:WPARameter<x>:HISTogram:
MEASure:CURSor:LINKage?
```
<x> = 1, 2

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:LINKAGE ON
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:LINKAGE? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:CURSOR:LINKAGE 1
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:MODE

Function Sets the automated measurement mode of the histogram display of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure:MODE {CURSor|OFF|PARameter}
:ANALysis:WPARameter<x>:HISTogram:
MEASure:MODE?
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:MODE CURSOR
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:MODE? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:MODE CURSOR
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:PARameter?

Function Queries all settings related to the automated measurement of histogram parameters of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure:PARameter?
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:MEAN:STATE 1;:
ANALYSIS:WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER:PEAK:STATE 1;:ANALYSIS:
WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER:SD3INTEG:STATE 1;:ANALYSIS:
WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER:SDEVIATION:STATE 1;:ANALYSIS:
WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER:SDINTEG:STATE 1
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:PARameter:ALL

Function Turn ON/OFF all histogram parameters of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>HISTogram:
MEASure:PARameter:ALL {<Boolean>}
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:ALL ON
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:PARameter:<Parameter>?

Function Queries all settings related to the histogram parameter of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure:PARameter:<Parameter>?
```
<x> = 1 or 2
<Parameter> = {MEAN|PEAK|SD3integ|SDEViation|SDINteg}

Example (The following is an example for the average value.)
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:MEAN? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:MEAN:STATE 1
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:PARameter:<Parameter>:STATe

Function Turns ON/OFF the histogram parameter of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure:PARameter:<Parameter>:STATe
{<Boolean>}
:ANALysis:WPARameter<x>:HISTogram:
MEASure:PARameter:<Parameter>:STATe?
```
<x> = 1 or 2
<Parameter> = {MEAN|PEAK|SD3integ|SDEViation|SDINteg}

Example (The following is an example for the average value.)
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:MEAN:STATE ON
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:MEAN:STATE? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:MEAN:STATE 1
```

:ANALysis:WPARameter<x>:HISTogram:MEASure:PARameter:<Parameter>:VALue?

Function Queries the measured value of the histogram parameter of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:HISTogram:
MEASure:PARameter:<Parameter>:VALue?
```
<x> = 1 or 2
<Parameter> = {MEAN|PEAK|SD3integ|SDEViation|SDINteg}

Example (The following is an example for the average value.)
```
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:MEAN:VALUE? ->
:ANALYSIS:WPARAMETER1:HISTOGRAM:
MEASURE:PARAMETER:MEAN:VALUE 1.000E+00
```

:ANALysis:WPARameter<x>:LIST?

Function Queries all settings related to the list display of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:LIST?
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:LIST? ->
:ANALYSIS:WPARAMETER1:LIST:
SCROLL HORIZONTAL
```

:ANALysis:WPARameter<x>:LIST:ITEM?

Function Queries list display items of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:LIST:ITEM?
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:LIST:ITEM? ->
:ANALYSIS:WPARAMETER1:LIST:
ITEM "LOW(C1)"
```

:ANALysis:WPARameter<x>:LIST:SCRoll

Function Sets the scroll direction of the list display of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:LIST:SCRoll
{HORizontal|VERTical}
:ANALysis:WPARameter<x>:LIST:SCRoll?
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:LIST:
SCROLL HORIZONTAL
:ANALYSIS:WPARAMETER1:LIST:SCROLL? ->
:ANALYSIS:WPARAMETER1:LIST:
SCROLL HORIZONTAL
```

:ANALysis:WPARameter<x>:LIST:VALue?

Function Queries the automated measured value of the list display number of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:LIST:
VALue? {<NRf>|MAXimum}
```
<x> = 1 or 2
<NRf> = 1 to 100000

Example
```
:ANALYSIS:WPARAMETER1:LIST:VALUE? 1 ->
:ANALYSIS:WPARAMETER1:LIST:
VALUE 1,1.000E+00
```

Description
- If the measurement is not possible, "NAN (Not A Number" is returned.
- If the value of the list display number is not present, "NAN (Not A Number)" is returned.
- If MAXimum is specified, the maximum list display number is selected.

:ANALysis:WPARameter<x>:MODE

Function Sets the mode of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:MODE
{HISTogram|LIST|TRENd}
:ANALysis:WPARameter<x>:MODE?
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:MODE HISTOGRAM
:ANALYSIS:WPARAMETER1:MODE? ->
:ANALYSIS:WPARAMETER1:MODE HISTOGRAM
```

:ANALysis:WPARameter<x>:TRACe<x>?

Function Queries all settings related to the trace of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:TRACe<x>?
```
<x> of WPARameter<x> = 1 or 2
<x> of TRACe<x> = 1 to 8

Example
```
:ANALYSIS:WPARAMETER1:TRACE1? ->
:ANALYSIS:WPARAMETER1:TRACE1:AREA1:
TYPE BURST
```

:ANALysis:WPARameter<x>:TRACe<x>: AREA<x>?

Function Queries all settings related to the area of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:TRACe<x>:
AREA<x>?
```
<x> of WPARameter<x> = 1 or 2
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:TRACE1:AREA1? ->
:ANALYSIS:WPARAMETER1:TRACE1:AREA1:
TYPE BURST
```

:ANALysis:WPARameter<x>:TRACe<x>: AREA<x>:TYPE

Function Sets the waveform parameter of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:TRACe<x>:
AREA<x>:TYPE {<Parameter>}
:ANALysis:WPARameter<x>:TRACe<x>:
AREA<x>:TYPE?
```
<x> of WPARameter<x> = 1 or 2
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2
<Parameter> = {BURSt|CMEan|COUNt|CRMS|CSDeviation|DELay|DUTYcycle|FALL|FREQuency|HIGH|HILow|LOW|MAXimum|MEAN|MINimum|NOVershoot|NWIDth|PERFrequency|PERiod|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEViation|TYCInteg|TYINteg}

Example
```
:ANALYSIS:WPARAMETER1:TRACE1:AREA1:
TYPE BURST
:ANALYSIS:WPARAMETER1:TRACE1:AREA1:
TYPE? -> :ANALYSIS:WPARAMETER1:TRACE1:
AREA1:TYPE BURST
```

:ANALysis:WPARameter<x>:TRENd?

Function Queries all settings related to the trend display of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:TRENd?
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:TREND? ->
:ANALYSIS:WPARAMETER1:TREND:
HSPAN 1;MEASURE:CURSOR:C1:
POSITION 1.000E+00;STATE 1;
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
CURSOR:C2:POSITION 2.000E+00;STATE 1;
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
CURSOR:DC:STATE 1;
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
MODE CURSOR;
:ANALYSIS:WPARAMETER1:TREND:
VERTICAL 2.000E+00,1.000E+00
```

:ANALysis:WPARameter<x>:TRENd:ASCale[:EXECute]

Function Executes the auto scaling of the trend display of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:
ASCale[:EXECute]
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:TREND:ASCALE:
EXECUTE
```

:ANALysis:WPARameter<x>:TRENd:HSPan

Function Sets the horizontal span of the trend display of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:HSPan
{<NRf>}
:ANALysis:WPARameter<x>:TRENd:HSPan?
```
<x> = 1 or 2
<NRf> = 1 to 100000

Example
```
:ANALYSIS:WPARAMETER1:TREND:HSPAN 1
:ANALYSIS:WPARAMETER1:TREND:HSPAN? ->
:ANALYSIS:WPARAMETER1:TREND:HSPAN 1
```

:ANALysis:WPARameter<x>:TRENd:MEASure?

Function Queries all settings related to the automated measurement of the trend display of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:MEASure?
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:TREND:MEASURE? ->
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
CURSOR:C1:POSITION 1.000E+00;STATE 1;
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
CURSOR:C2:POSITION 2.000E+00;STATE 1;
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
CURSOR:DC:STATE 1;
:ANALYSIS:WPARAMETER1:TREND:
MEASURE:MODE CURSOR
```

:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor?

Function Queries all settings related to the cursor measurement of the trend of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:MEASure:
CURSor?
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
CURSOR? -> :ANALYSIS:WPARAMETER1:TREND:
MEASURE:CURSOR:C1:POSITION 1.000E+00;
STATE 1;:ANALYSIS:WPARAMETER1:TREND:
MEASURE:CURSOR:C2:POSITION 2.000E+00;
STATE 1;:ANALYSIS:WPARAMETER1:TREND:
MEASURE:CURSOR:DC:STATE 1
```

:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:ALL

Function Turns ON/OFF all cursors of the trend of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:MEASure:
CURSor:ALL {<Boolean>}
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
CURSOR:ALL ON
```

:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:C<x>?

Function Queries all settings related to the measured value of the cursor of the trend of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:MEASure:
CURSor:C<x>?
```
<x> of WPARameter<x> = 1 or 2
<x> of C<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
CURSOR:C1? -> :ANALYSIS:WPARAMETER1:
TREND:MEASURE:CURSOR:C1:
POSITION 1.000E+00;STATE 1
```

:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:C<x>:POSition

Function Sets the cursor position of the trend of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:MEASure:
CURSor:C<x>:POSition {<NRf>}
:ANALysis:WPARameter<x>:TRENd:MEASure:
CURSor:C<x>:POSition?
```
<x> of WPARameter<x> = 1 or 2
<x> of C<x> = 1 or 2
<NRf> = –5 to 5 div

Example
```
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
CURSOR:C1:POSITION 1
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
CURSOR:C1:POSITION? ->
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
CURSOR:C1:POSITION 1.000E+00
CURSOR:C1:STATE? ->
:ANALYSIS:WPARAMETER1:TREND:MEASURE:
CURSOR:C1:STATE 1
```

:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:C<x>:STATe

Function Turns ON/OFF the cursor of the trend of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:C<x>:STATe {<Boolean>}
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:C<x>:STATe?
```
<x> of WPARameter<x> = 1 or 2
<x> of C<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:C1:STATE ON
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:C1:STATE? ->
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:C1:STATE 1
```

:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:C<x>:VALue?

Function Queries the measured value of the cursor of the trend of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:C<x>:VALue?
```
<x> of WPARameter<x> = 1 or 2
<x> of C<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:C1:VALUE? ->
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:C1:VALUE 1.000E+00
```

:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:DC?

Function Queries all settings related to the measurement between cursors on the trend of the waveform parameter measurement.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:DC?
```
<x> of WPARameter<x> = 1, 2

Example
```
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:DC? ->
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:DC:STATE 1
```

:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:DC:STATe

Function Turns ON/OFF the measurement between cursors on the trend of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:DC:STATe {<Boolean>}
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:DC:STATe?
```
<x> of WPARameter<x> = 1, 2

Example
```
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:DC:STATE ON
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:DC:STATE? ->
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:DC:STATE 1
```

:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:DC:VALue?

Function Sets the measured value between cursors on the trend of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:DC:VALue?
```
<x> of WPARameter<x> = 1, 2

Example
```
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:DC:VALUE? ->
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:DC:VALUE 1.000E+00
```

:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:LINKage

Function Turns ON/OFF the trend cursor link of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:LINKage {<Boolean>}
:ANALysis:WPARameter<x>:TRENd:MEASure:CURSor:LINKage?
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:LINKAGE ON
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:LINKAGE? ->
:ANALYSIS:WPARAMETER1:TREND:MEASURE:CURSOR:LINKAGE 1
```

:ANALysis:WPARameter<x>:TRENd:MEASure:MODE

Function Sets the automated measurement mode of the trend of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:MEASure:MODE {CURSor|OFF}
:ANALysis:WPARameter<x>:TRENd:MEASure:MODE?
```
<x> = 1 or 2

Example
```
:ANALYSIS:WPARAMETER1:TREND:MEASURE:MODE CURSOR
:ANALYSIS:WPARAMETER1:TREND:MEASURE:MODE? -> :ANALYSIS:WPARAMETER1:TREND:MEASURE:MODE CURSOR
```

:ANALysis:WPARameter<x>:TRENd:VERTical

Function Sets the vertical range of the trend of the waveform parameter measurement or queries the current setting.

Syntax
```
:ANALysis:WPARameter<x>:TRENd:VERTical {<NRf>,<NRf>}
:ANALysis:WPARameter<x>:TRENd:VERTical?
```
<x> = 1 or 2
<NRf> = –1.000E+31 to 1.000E+31 (div)

Example
```
:ANALYSIS:WPARAMETER1:TREND:VERTICAL 1,2
:ANALYSIS:WPARAMETER1:TREND:VERTICAL? -> :ANALYSIS:WPARAMETER1:TREND:VERTICAL 2.000E+00,1.000E+00
```

:ANALysis:XY<x>?

Function Queries all settings related to the XY display function.

Syntax
```
:ANALysis:XY<x>?
```
<x> = 1 or 2

Example
```
:ANALYSIS:XY1? -> :ANALYSIS:XY1:GATE:ALEVEL HIGH;HYSTERESIS1 1.000E+00;
HYSTERESIS2 1.000E+00;
HYSTERESIS3 1.000E+00;
HYSTERESIS4 1.000E+00;
HYSTERESIS5 1.000E+00;
HYSTERESIS6 1.000E+00;
HYSTERESIS7 1.000E+00;
HYSTERESIS8 1.000E+00;LEVEL1 1.000E+00;
LEVEL2 1.000E+00;LEVEL3 1.000E+00;
LEVEL4 1.000E+00;LEVEL5 1.000E+00;
LEVEL6 1.000E+00;LEVEL7 1.000E+00;
LEVEL8 1.000E+00;TRACE 1;:ANALYSIS:XY1:MEASURE:CURSOR:X1:POSITION 1.000E+00;:ANALYSIS:XY1:MEASURE:CURSOR:X2:POSITION 2.000E+00;:ANALYSIS:XY1:MEASURE:CURSOR:XLINkAGE 1;Y1:POSITION 1.000E+00;:ANALYSIS:XY1:MEASURE:CURSOR:Y2:POSITION 2.000E+00;:ANALYSIS:XY1:MEASURE:CURSOR:YLINkAGE 1;:ANALYSIS:XY1:MEASURE:MODE CURSOR;XYINTEG:LOOP CLOSE;POLARITY CCW;:ANALYSIS:XY1:TRANGE 1.000E+00,0.000E+00;WINDOW MAIN;XTRACE 1;YTRACE 1
```

:ANALysis:XY<x>:GATE?

Function Queries all settings related to the gate function of the XY display.

Syntax
```
:ANALysis:XY<x>:GATE?
```
<x> = 1 or 2

Example
```
:ANALYSIS:XY1:GATE? ->
:ANALYSIS:XY1:GATE:ALEVEL HIGH;
HYSTERESIS1 1.000E+00;
HYSTERESIS2 1.000E+00;
HYSTERESIS3 1.000E+00;
HYSTERESIS4 1.000E+00;
HYSTERESIS5 1.000E+00;
HYSTERESIS6 1.000E+00;
HYSTERESIS7 1.000E+00;
HYSTERESIS8 1.000E+00;LEVEL1 1.000E+00;
LEVEL2 1.000E+00;LEVEL3 1.000E+00;
LEVEL4 1.000E+00;LEVEL5 1.000E+00;
LEVEL6 1.000E+00;LEVEL7 1.000E+00;
LEVEL8 1.000E+00;TRACE 1
```

:ANALysis:XY<x>:GATE:ALEVel

Function Sets the active level of the gate of the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:GATE:ALEVel {HIGH|LOW}
:ANALysis:XY<x>:GATE:ALEVel?
```
<x> = 1 or 2

Example
```
:ANALYSIS:XY1:GATE:ALEVEL HIGH
:ANALYSIS:XY1:GATE:ALEVEL? ->
:ANALYSIS:XY1:GATE:ALEVEL HIGH
```

:ANALysis:XY<x>:GATE:HYSTeresis<x>

Function Sets the hysteresis of the gate of the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:GATE:HYSTeresis<x>
{<NRf>}
:ANALysis:XY<x>:GATE:HYSTeresis<x>?
```
<x> of XY<x> = 1 or 2
<x> of HYSTeresis<x> = 1 to 8
<NRf> = 0 to 4 (div)

Example
```
:ANALYSIS:XY1:GATE:HYSTERESIS1 1
:ANALYSIS:XY1:GATE:HYSTERESIS1? ->
:ANALYSIS:XY1:GATE:
HYSTERESIS1 1.000E+00
```

:ANALysis:XY<x>:GATE:LEVel<x>

Function Sets the level of the gate of the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:GATE:LEVel<x>
{<NRf>,<Voltage>,<Current>}
:ANALysis:XY<x>:GATE:LEVel<x>?
```
<x> of XY<x> = 1 or 2
<x> of LEVel<x> = 1 to 8
<NRf>, <Voltage>, and <Current> = See the DL9000 User's Manual.

Example
```
:ANALYSIS:XY1:GATE:LEVEL1 1
:ANALYSIS:XY1:GATE:LEVEL1? ->
:ANALYSIS:XY1:GATE:LEVEL1 1.000E+00
```

:ANALysis:XY<x>:GATE:TRACe

Function Sets the gate trace of the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:GATE:TRACe {<NRf>|OFF}
:ANALysis:XY<x>:GATE:TRACe?
```
<x> = 1 or 2
<NRf> = 1 to 8

Example
```
:ANALYSIS:XY1:GATE:TRACE 1
:ANALYSIS:XY1:GATE:TRACE? ->
:ANALYSIS:XY1:GATE:TRACE 1
```

Function Queries all settings related to the automated measurement of the XY display.

Syntax
```
:ANALysis:XY<x>:MEASure?
```
<x> = 1 or 2

Example
```
:ANALYSIS:XY1:MEASURE? -> :ANALYSIS:
XY1:MEASURE:CURSOR:X1:
POSITION 1.000E+00;:ANALYSIS:XY1:
MEASURE:CURSOR:X2:POSITION 2.000E+00;:
ANALYSIS:XY1:MEASURE:CURSOR:XLINkAGE 1;
Y1:POSITION 1.000E+00;:ANALYSIS:XY1:
MEASURE:CURSOR:Y2:POSITION 2.000E+00;:
ANALYSIS:XY1:MEASURE:CURSOR:
YLINkAGE 1;:ANALYSIS:XY1:MEASURE:
MODE CURSOR;XYINTEG:LOOP CLOSE;
POLARITY CCW
```

:ANALysis:XY<x>:MEASure?

Function Queries all settings related to the automated measurement of the XY display.

Syntax
```
:ANALysis:XY<x>:MEASure?
```
<x> = 1 or 2

Example
```
:ANALYSIS:XY1:MEASURE? -> :ANALYSIS:
XY1:MEASURE:CURSOR:X1:
POSITION 1.000E+00;:ANALYSIS:XY1:
MEASURE:CURSOR:X2:POSITION 2.000E+00;:
ANALYSIS:XY1:MEASURE:CURSOR:XLINkAGE 1;
Y1:POSITION 1.000E+00;:ANALYSIS:XY1:
MEASURE:CURSOR:Y2:POSITION 2.000E+00;:
ANALYSIS:XY1:MEASURE:CURSOR:
YLINkAGE 1;:ANALYSIS:XY1:MEASURE:
MODE CURSOR;XYINTEG:LOOP CLOSE;
POLARITY CCW
```

:ANALysis:XY<x>:MEASure:CURSor?

Function Queries all settings related to the cursor measurement of the XY display.

Syntax
```
:ANALysis:XY<x>:MEASure:CURSor?
```
<x> = 1 or 2

Example
```
:ANALYSIS:XY1:MEASURE:CURSOR? ->
:ANALYSIS:XY1:MEASURE:CURSOR:X1:
POSITION 1.000E+00;:ANALYSIS:XY1:
MEASURE:CURSOR:X2:POSITION 2.000E+00;:
ANALYSIS:XY1:MEASURE:CURSOR:XLINkAGE 1;
Y1:POSITION 1.000E+00;:ANALYSIS:XY1:
MEASURE:CURSOR:Y2:POSITION 2.000E+00;:
ANALYSIS:XY1:MEASURE:CURSOR:YLINkAGE 1
```

:ANALysis:XY<x>:MEASure:CURSor:XLINkage

Function Turns ON/OFF the horizontal cursor link on the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:MEASure:CURSor:XLINkage {<Boolean>}
:ANALysis:XY<x>:MEASure:CURSor:XLINkage?
```
<x> = 1 or 2

Example
```
:ANALYSIS:XY1:MEASURE:CURSOR:XLINkAGE ON
:ANALYSIS:XY1:MEASURE:CURSOR:XLINkAGE?
-> :ANALYSIS:XY1:MEASURE:CURSOR:XLINkAGE 1
```

:ANALysis:XY<x>:MEASure:CURSor:X<x>?

Function Queries all settings related to the horizontal cursor of the XY display.

Syntax
```
:ANALysis:XY<x>:MEASure:CURSor:X<x>?
```
<x> of XY<x> = 1 or 2
<x> of X<x> = 1 or 2

Example
```
:ANALYSIS:XY1:MEASURE:CURSOR:X1? ->
:ANALYSIS:XY1:MEASURE:CURSOR:X1:POSITION 1.000E+00
```

:ANALysis:XY<x>:MEASure:CURSor:Y<x>:POSition

Function Sets the vertical cursor position of the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:MEASure:CURSor:Y<x>:POSition {<NRf>}
:ANALysis:XY<x>:MEASure:CURSor:Y<x>:POSition?
```
<x> of XY<x> = 1 or 2
<x> of Y<x> = 1 or 2
<NRf> = –4 to 4 (div)

Example
```
:ANALYSIS:XY1:MEASURE:CURSOR:Y1:POSITION 1
:ANALYSIS:XY1:MEASURE:CURSOR:Y1:POSITION? -> :ANALYSIS:XY1:MEASURE:CURSOR:Y1:POSITION 1.000E+00
```

:ANALysis:XY<x>:MEASure:CURSor:X<x>:VALue?

Function Queries the voltage value at the horizontal cursor of the XY display.

Syntax
```
:ANALysis:XY<x>:MEASure:CURSor:X<x>:VALue?
```
<x> of XY<x> = 1 or 2
<x> of X<x> = 1 or 2

Example
```
:ANALYSIS:XY1:MEASURE:CURSOR:X1:VALUE?
-> :ANALYSIS:XY1:MEASURE:CURSOR:X1:VALUE 1.000E+00
```

:ANALysis:XY<x>:MEASure:CURSor:YLINkage

Function Turns ON/OFF the vertical cursor link on the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:MEASure:CURSor:YLINkage {<Boolean>}
:ANALysis:XY<x>:MEASure:CURSor:YLINkage?
```
<x> = 1 or 2

Example
```
:ANALYSIS:XY1:MEASURE:CURSOR:YLINkAGE ON
:ANALYSIS:XY1:MEASURE:CURSOR:YLINkAGE?
-> :ANALYSIS:XY1:MEASURE:CURSOR:YLINkAGE 1
```

:ANALysis:XY<x>:MEASure:CURSor:Y<x>?

Function Queries all settings related to the vertical cursor of the XY display.

Syntax
```
:ANALysis:XY<x>:MEASure:CURSor:Y<x>?
```
<x> of XY<x> = 1 or 2
<x> of Y<x> = 1 or 2

Example
```
:ANALYSIS:XY1:MEASURE:CURSOR:Y1? ->
:ANALYSIS:XY1:MEASURE:CURSOR:Y1:POSITION 1.000E+00
```

:ANALysis:XY<x>:MEASure:CURSor:Y<x>:POSition

Function Sets the vertical cursor position of the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:MEASure:CURSor:Y<x>:POSition {<NRf>}
:ANALysis:XY<x>:MEASure:CURSor:Y<x>:POSition?
```
<x> of XY<x> = 1 or 2
<x> of Y<x> = 1 or 2
<NRf> = –4 to 4 (div)

Example
```
:ANALYSIS:XY1:MEASURE:CURSOR:Y1:POSITION 1
:ANALYSIS:XY1:MEASURE:CURSOR:Y1:POSITION? -> :ANALYSIS:XY1:MEASURE:CURSOR:Y1:POSITION 1.000E+00
```

:ANALysis:XY<x>:MEASure:CURSor:Y<x>:VALue?

Function Queries the voltage value at the vertical cursor of the XY display.

Syntax
```
:ANALysis:XY<x>:MEASure:CURSor:Y<x>:
VALue?
```
<x> of XY<x> = 1 or 2
<x> of Y<x> = 1 or 2

Example
```
:ANALYSIS:XY1:MEASURE:CURSOR:Y1:VALUE?
-> :ANALYSIS:XY1:MEASURE:CURSOR:Y1:
VALUE 1.000E+00
```

:ANALysis:XY<x>:MEASure:MODE

Function Sets the automated measurement mode of the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:MEASure:MODE {CURSor|
OFF|XYINteg}
:ANALysis:XY<x>:MEASure:MODE?
```
<x> = 1 or 2

Example
```
:ANALYSIS:XY1:MEASURE:MODE CURSOR
:ANALYSIS:XY1:MEASURE:MODE? ->
:ANALYSIS:XY1:MEASURE:MODE CURSOR
```

:ANALysis:XY<x>:MEASure:XYINteg?

Function Queries all settings related to the integration of the XY display.

Syntax
```
:ANALysis:XY<x>:MEASure:XYINteg?
```
<x> = 1 or 2

Example
```
:ANALYSIS:XY1:MEASURE:XYINTEG? ->
:ANALYSIS:XY1:MEASURE:XYINTEG:
LOOP CLOSE;POLARITY CCW
```

:ANALysis:XY<x>:MEASure:XYINteg:LOOP

Function Sets the integration mode of the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:MEASure:XYINteg:LOOP
{CLOSe|OPEN}
:ANALysis:XY<x>:MEASure:XYINteg:LOOP?
```
<x> = 1 or 2

Example
```
:ANALYSIS:XY1:MEASURE:XYINTEG:
LOOP CLOSE
:ANALYSIS:XY1:MEASURE:XYINTEG:LOOP? ->
:ANALYSIS:XY1:MEASURE:XYINTEG:
LOOP CLOSE
```

:ANALysis:XY<x>:MEASure:XYINteg:POLarity

Function Sets the integration polarity of the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:MEASure:XYINteg:
POLarity {CCW|CW}
:ANALysis:XY<x>:MEASure:XYINteg:
POLarity?
```
<x> = 1 or 2

Example
```
:ANALYSIS:XY1:MEASURE:XYINTEG:
POLARITY CCW
:ANALYSIS:XY1:MEASURE:XYINTEG:POLARITY?
-> :ANALYSIS:XY1:MEASURE:XYINTEG:
POLARITY CCW
```

:ANALysis:XY<x>:MEASure:XYINteg:VALue?

Function Queries the integral value of the XY display.

Syntax
```
:ANALysis:XY<x>:MEASure:XYINteg:VALue?
```
<x> = 1 or 2

Example
```
:ANALYSIS:XY1:MEASURE:XYINTEG:VALUE? ->
:ANALYSIS:XY1:MEASURE:XYINTEG:
VALUE 1.000E+00
```

:ANALysis:XY<x>:TRANge (Time Range)

Function Sets the measurement range of the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:TRANge {<NRf>,<NRf>}
:ANALysis:XY<x>:TRANge?
```
<x> = 1 or 2
<NRf> = –5 to 5 div

:ANALysis:XY<x>:WINDow

Function Sets the measurement source window of the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:WINDow {MAIN|Z1|Z2}
:ANALysis:XY<x>:WINDow?
```
<x> = 1 or 2

Example
```
:ANALYSIS:XY1:WINDOW MAIN
:ANALYSIS:XY1:WINDOW? ->
:ANALYSIS:XY1:WINDOW MAIN
```

:ANALysis:XY<x>:XTRace

Function Sets the X-axis trace of the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:XTRace {<NRf>}
:ANALysis:XY<x>:XTRace?
```
<x> = 1 or 2
<NRf> = 1 to 8

Example
```
:ANALYSIS:XY1:XTRACE 1
:ANALYSIS:XY1:XTRACE? ->
:ANALYSIS:XY1:XTRACE 1
```

:ANALysis:XY<x>:YTRace

Function Sets the Y-axis trace of the XY display or queries the current setting.

Syntax
```
:ANALysis:XY<x>:YTRace {<NRf>}
:ANALysis:XY<x>:YTRace?
```
<x> = 1 or 2
<NRf> = 1 to 8

Example
```
:ANALYSIS:XY1:YTRACE 1
:ANALYSIS:XY1:YTRACE? ->
:ANALYSIS:XY1:YTRACE 1
```

5.4 ASETup Group

:ASETup:EXECute

Function Execute auto setup.

Syntax
```
:ASETup:EXECute
```

Example
```
:ASETUP:EXECUTE
```

:ASETup:UNDO

Function Cancels auto setup that has been executed.

Syntax
```
:ASETup:UNDO
```

Example
```
:ASETUP:UNDO
```

5.5 CALibrate Group

:CALibrate?

Function Queries all settings related to the calibration.

Syntax
```
:CALibrate?
```

Example
```
:CALIBRATE? -> :CALIBRATE:MODE AUTO
```

:CALibrate:EXECute

Function Executes 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:MODE?
```

Example
```
:CALIBRATE:MODE AUTO
:CALIBRATE:MODE? ->
:CALIBRATE:MODE AUTO
```

5.6 CHANnel Group

:CHANnel<x>?

Function Queries all settings related to the channel.
Syntax :CHANnel<x>?
<x> = 1 to 4
Example :CHANNEL1? -> :CHANNEL1:SELECT INPUT;
DISPLAY 1;BWIDTH FULL;COUPLING DC;
DESKEW 0.000E+00;INVERT 0;LABEL:
DEFINE “CH1”;MODE 1;:CHANNEL1:
OCANCEL 0;OFFSET 0.000E+00;
POSITION 0.000E+00;PROBE:MODE 1;:
CHANNEL1:SVALUE 0;VDIV 1.000E+00

:CHANnel<x>:ASCale[:EXECute]

Function Executes the auto scaling of the channel.
Syntax CHANnel<x>:ASCale[:EXECute]
<x> = 1 to 4
Example CHANNEL1:ASCALE:EXECUTE

:CHANnel<x>:BWIDth

Function Sets the input filter of the channel or queries the current setting.
Syntax :CHANneL<x>:BWIDth {<Frequency>|FULL}
:CHANneL<x>:BWIDth?
<x> = 1 to 4
<Frequency> = See the DL9000 User’s Manual
Example :CHANNEL1:BWIDTH FULL
:CHANNEL1:BWIDTH? -> :CHANNEL1:BWIDTH
FULL

:CHANnel<x>:COUPling

Function Sets the input coupling of the channel or queries the current setting.
Syntax :CHANneL<x>:COUPling {AC|DC|DC50|GND}
:CHANneL<x>:COUPling?
<x> = 1 to 4
Example :CHANNEL1:COUPLING GND
:CHANNEL1:COUPLING? ->
:CHANNEL1:COUPLING GND

:CHANnel<x>:DESKew

Function Sets the skew correction of the channel or queries the current setting.
Syntax :CHANneL<x>:DESKew {<Time>}
:CHANneL<x>:DESKew?
<x> = 1 to 4
<Time> = –100 ns to 100 ns (10 ps steps)
Example :CHANNEL1:DESKEW 1NS
:CHANNEL1:DESKEW? ->
:CHANNEL1:DESKEW 1.000E-09

:CHANnel<x>:DISPlay

Function Turns ON/OFF the display of the channel or queries the current setting.
Syntax :CHANneL<x>:DISPlay {<Boolean>}
:CHANneL<x>:DISPlay?
<x> = 1 to 4
Example :CHANNEL1:DISPLAY ON
:CHANNEL1:DISPLAY? ->
:CHANNEL1:DISPLAY 1

:CHANnel<x>:INVert

Function Turns ON/OFF the inverted display of the channel or queries the current setting.
Syntax :CHANneL<x>:INVert {<Boolean>}
:CHANneL<x>:INVert?
<x> = 1 to 4
Example :CHANNEL1:INVERT ON
:CHANNEL1:INVERT -> :CHANNEL1:INVERT 1

:CHANnel<x>:LABel?

Function Queries all settings related to the waveform label of the channel.
Syntax :CHANneL<x>:LABel?
<x> = 1 to 4
Example :CHANNEL1:LABEL? ->
:CHANNEL1:LABEL:DEFINE “CH1”;MODE 0

:CHANnel<x>:LABel[:DEFine]

Function Sets the waveform label of channel or queries the current setting.
Syntax :CHANneL<x>:LABel[:DEFine] {<String>}
:CHANneL<x>:LABel[:DEFine]?
<x> = 1 to 4
<String> = Up to 8 characters
Example :CHANNEL1:LABEL:DEFINE “CH1”
:CHANNEL1:LABEL:DEFINE? ->
:CHANNEL1:LABEL:DEFINE “CH1”

:CHANnel<x>:LABel:MODE

Function Turns ON/OFF the waveform label display of the channel or queries the current setting.
Syntax :CHANneL<x>:LABel:MODE {<Boolean>}
:CHANneL<x>:LABel:MODE?
<x> = 1 to 4
Example :CHANNEL1:LABEL:MODE ON
:CHANNEL1:LABEL:MODE? ->
:CHANNEL1:LABEL:MODE 1

:CHANnel<x>:OCANcel

Function Turns ON/OFF the offset cancel of the channel or queries the current setting.

Syntax
```
:CHANneL<x>:OCANcel {<Boolean>}
:CHANneL<x>:OCANcel?
```
<x> = 1 to 4

Example
```
:CHANNEL1:OCANCEL ON
:CHANNELL1:OCANCEL? ->
:CHANNEL1:OCANCEL 1
```

:CHANnel<x>:OFFSet

Function Sets the offset voltage of the channel or queries the current setting.

Syntax
```
:CHANneL<x>:OFFSet
{<Voltage>|<Current>}
:CHANneL<x>:OFFSet?
```
<x> = 1 to 4
<Voltage> and <Current> = See the DL9000 User's Manual.

Example
```
:CHANNEL1:OFFSET 1V
:CHANNEL1:OFFSET? ->
:CHANNEL1:OFFSET 1.000E+00
```

:CHANnel<x>:POSition

Function Sets the vertical position of the channel or queries the current setting.

Syntax
```
:CHANneL<x>:POSition {<NRf>}
:CHANneL<x>:POSition?
```
<x> = 1 to 4
<NRf> = –4 to 4 (div)

Example
```
:CHANNEL1:POSITION 1
:CHANNEL1:POSITION? ->
:CHANNEL1:POSITION 1.000E+00
```

:CHANnel<x>:PROBe?

Function Queries all settings related to the probe attenuation of the channel.

Syntax
```
:CHANneL<x>:PROBe?
```
<x> = 1 to 4

Example
```
:CHANNEL1:PROBE? ->
:CHANNEL1:PROBE:MODE 1
```

:CHANnel<x>:PROBe[:MODE]

Function Sets the probe attenuation of the channel or queries the current setting.

Syntax
```
:CHANneL<x>:PROBe[:MODE] {<NRf>|AUTO|
C10|C100}
:CHANneL<x>:PROBe[:MODE]?
```
<x> = 1 to 4
<NRf> = 1,2,5,10,20,50,100,200,500,1000

Example
```
:CHANNEL1:PROBE:MODE 1
:CHANNEL1:PROBE:MODE? ->
:CHANNEL1:PROBE:MODE 1
```

:Channal<x>:PROBe:AUTO?

Function Queries the probe attenuation of the channel when set to AUTO.

Syntax
```
:Channal<x>:PROBe:AUTO?
```
<x> = 1 to 4

Example
```
:CHANNEL1:PROBE:AUTO? ->
:CHANNEL1:PROBE:AUTO 1
```

:CHANnel<x>:SELect

Function Sets the waveform (input/computation) to be assigned to the input channel or queries the current setting.

Syntax
```
:CHANnel<x>:SELect {INPut|MATH}
:CHANnel<x>:SELect?
```
<x> = 1 to 4

Example
```
:CHANNEL1:SELECT INPUT
:CHANNEL1:SELECT? -> :CHANNEL1:SELECT
INPUT
```

:CHANnel<x>:SVALue (Scale VALUE)

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

Syntax
```
:CHANnel<x>:SVALue {<Boolean>}
:CHANnel<x>:SVALue?
```
<x> = 1 to 4

Example
```
:CHANNEL1:SVALUE ON
:CHANNEL1:SVALUE? -> :CHANNEL1:SVALUE 1
```

:CHANnel<x>:VDIV

Function Sets the vertical sensitivity (V/div) of the channel or queries the current setting.

Syntax
```
:CHANnel<x>:VDIV {<Voltage>|<Current>}
:CHANnel<x>:VDIV?
```
<x> = 1 to 4
<Voltage> and <Current> = See the DL9000 User's Manual.

Example
```
:CHANNEL1:VDIV 5V
:CHANNEL1:VDIV? ->
:CHANNEL1:VDIV 5.000E+00
```

5.7 CLEar Group

:CLEar:ACCumulate

Function Clears accumulated waveforms.
Syntax :CLEar:ACCumulate
Example :CLEAR:ACCUMULATE

:CLEar[:HISTory]

Function Clears history waveforms.
Syntax :CLEar[:HISTory]
Example :CLEAR:HISTORY

:CLEar:SNAP

Function Clears snapshot waveforms.
Syntax :CLEar:SNAP
Example :CLEAR:SNAP

5.8 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) or queries the current setting.
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 dedicated to the Ethernet interface (option).

:COMMunicate:OPSE (Operation Pending Status Enable register)

Function Sets the overlap command that is 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 2400
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, 8, and 11 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 2400
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, 8, and 11 only.
- For the description of how to synchronize the program using :COMMunicate:OVERlap, see page 4-7.
- In the above example, bits 5, 6, 8, and 11 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. ON is remote mode.
Syntax :COMMunicate:REMote {<Boolean>}
:COMMunicate:REMote?
Example :COMMUNICATE:REMOTE ON
:COMMUNICATE:REMOTE? ->
:COMMUNICATE:REMOTE 1
Description This command is dedicated to the Ethernet interface (option).

: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
0	Unrecoverable transmission error
1	Always 0
2	Always 0
3 or greater	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`) or queries the current setting.

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 6-4.)

Example `:COMMUNICATE:WAIT 65535`

Description For the description of how to synchronize the program using `:COMMunicate:WAIT`, see page 3-8.

: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 6-4.)

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	SCH	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

When bit 11 (SCH) = 1:
Search execution not complete

5.9 CURSor Group

:CURSor?

Function Queries all settings related to the cursor measurement.

Syntax :CURSor?

Example :CURSOR? -> :CURSOR:DISPLAY 1;
HORIZONTAL:BASIC:DV:STATE 1;:CURSOR:
HORIZONTAL:BASIC:LINKAGE 1;V1:
POSITION 1.000E+00;STATE 0;:CURSOR:
HORIZONTAL:BASIC:V2:POSITION 1.000E+00;
STATE 0;:CURSOR:HORIZONTAL:CALCULATION:
DEFINE1 "V1";DEFINE2 "V2";DEFINE3 "V1";
DEFINE4 "V2";STATE1 0;STATE2 0;
STATE3 0;STATE4 0;:CURSOR:HORIZONTAL:
TRACE 1;:CURSOR:MARKER:CALCULATION:
DEFINE1 "T2";DEFINE2 "V2";DEFINE3 "V3";
DEFINE4 "V4";STATE1 0;STATE2 0;
STATE3 0;STATE4 0;:CURSOR:MARKER:CM1:
DT2:STATE 1;:CURSOR:MARKER:CM1:DT3:
STATE 1;:CURSOR:MARKER:CM1:DT4:
STATE 1;:CURSOR:MARKER:CM1:DV2:
STATE 1;:CURSOR:MARKER:CM1:DV3:
STATE 1;:CURSOR:MARKER:CM1:DV4:
STATE 1;:CURSOR:MARKER:CM1:
POSITION 1.000E+00;T:STATE 1;:CURSOR:
MARKER:CM1:TRACE 1;V:STATE 1;:CURSOR:
MARKER:CM2:DT1:STATE 1;:CURSOR:MARKER:
CM2:DT3:STATE 1;:CURSOR:MARKER:CM2:DT4:
STATE 1;:CURSOR:MARKER:CM2:DV1:
STATE 1;:CURSOR:MARKER:CM2:DV3:
STATE ......

:CURSor:DISPlay

Function Turns ON/OFF the cursor or queries the current setting.

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

Example :CURSOR:DISPLAY ON
:CURSOR:DISPLAY? -> :CURSOR:DISPLAY 1

:CURSor:HORizontal?

Function Queries all settings related to the horizontal cursors.

Syntax :CURSor:HORizontal?

Example :CURSOR:HORIZONTAL? ->
:CURSOR:HORIZONTAL:BASIC:DV:STATE 1;:
CURSOR:HORIZONTAL:BASIC:LINKAGE 1;V1:
POSITION 1.000E+00;STATE 0;:CURSOR:
HORIZONTAL:BASIC:V2:POSITION 1.000E+00;
STATE 0;:CURSOR:HORIZONTAL:
CALCULATION:DEFINE1 "V1";DEFINE2 "V2";
DEFINE3 "V1";DEFINE4 "V2";STATE1 0;
STATE2 0;STATE3 0;STATE4 0;:CURSOR:
HORIZONTAL:TRACE 1

:CURSor:HORizontal:BASic?

Function Queries all settings related to basic items of the horizontal cursors.

Syntax :CURSor:HORizontal:BASic?

Example :CURSOR:HORIZONTAL:BASIC? ->
:CURSOR:HORIZONTAL:BASIC:DV:STATE 1;:
CURSOR:HORIZONTAL:BASIC:LINKAGE 1;V1:
POSITION 1.000E+00;STATE 0;:CURSOR:
HORIZONTAL:BASIC:V2:POSITION 1.000E+00;
STATE 0

:CURSor:HORizontal[:BASic]:ALL

Function Turns ON/OFF all basic items of the horizontal cursors.

Syntax :CURSor:HORizontal[:BASic]:ALL
{<Boolean>}

Example :CURSOR:HORIZONTAL:BASIC:ALL ON

:CURSor:HORizontal[:BASic]:DV?

Function Queries all settings related to the ΔV measurement of the horizontal cursors.

Syntax :CURSor:HORizontal[:BASic]:DV?

Example :CURSOR:HORIZONTAL:BASIC:DV? ->
:CURSOR:HORIZONTAL:BASIC:DV:STATE 1

:CURSor:HORizontal[:BASic]:DV:STATe

Function Turns ON/OFF the ΔV measurement of the horizontal cursors or queries the current setting.

Syntax :CURSor:HORizontal[:BASic]:DV:STATe
{<Boolean>}
:CURSor:HORizontal[:BASic]:DV:STATe?

Example :CURSOR:HORIZONTAL:BASIC:DV:STATE ON
:CURSOR:HORIZONTAL:BASIC:DV:STATE? ->
:CURSOR:HORIZONTAL:BASIC:DV:STATE 1

:CURSor:HORizontal[:BASic]:DV:VALue?

Function Queries the voltage value between the horizontal cursors.

Syntax :CURSor:HORizontal[:BASic]:DV:VALue?

Example :CURSOR:HORIZONTAL:BASIC:DV:VALUE? ->
:CURSOR:HORIZONTAL:BASIC:DV:
VALUE 1.000E+00

:CURSor:HORizontal[:BASic]:LINKage

Function Turns ON/OFF the horizontal cursor link or queries the current setting.

Syntax :CURSor:HORizontal[:BASic]:
LINKage {<Boolean>}
:CURSor:HORizontal[:BASic]:LINKage?

Example :CURSOR:HORIZONTAL:BASIC:LINKAGE ON
:CURSOR:HORIZONTAL:BASIC:LINKAGE? ->
:CURSOR:HORIZONTAL:BASIC:LINKAGE 1

:CURSor:HORizontal[:BASic]:V<x>?

Function Queries all settings related to the horizontal cursor.

Syntax
```
:CURSor:HORizontal[:BASic]:V<x>?
```
<x> = 1 or 2

Example
```
:CURSOR:HORIZONTAL:BASIC:V1? ->
:CURSOR:HORIZONTAL:BASIC:V1:POSITION
1.000E+00;STATE 1
```

:CURSor:HORizontal[:BASic]:V<x>:JUMP

Function Jumps the horizontal cursor to the center position of the zoom waveform.

Syntax
```
:CURSor:HORizontal[:BASic]:V<x>:
JUMP {Z1|Z2}
```
<x>=1 or 2

Example
```
:CURSOR:HORIZONTAL:BASIC:V1:JUMP Z1
```

:CURSor:HORizontal[:BASic]:V<x>:POSition

Function Sets the horizontal cursor position or queries the current setting.

Syntax
```
:CURSor:HORizontal[:BASic]:V<x>:
POSition {<NRf>}
:CURSor:HORizontal[:BASic]:V<x>:
POSition?
```
<x> = 1 or 2
<NRf> = –4 to 4 (div)

Example
```
:CURSOR:HORIZONTAL:BASIC:V1:POSITION 1
:CURSOR:HORIZONTAL:BASIC:V1:POSITION?
-> :CURSOR:HORIZONTAL:BASIC:V1:
POSITION 1.000E+00
```

:CURSor:HORizontal[:BASic]:V<x>:STATe

Function Turns ON/OFF the horizontal cursor or queries the current setting.

Syntax
```
:CURSor:HORizontal[:BASic]:V<x>:STATe
{<Boolean>}
:CURSor:HORizontal[:BASic]:V<x>:STATe?
```
<x> = 1 or 2

Example
```
:CURSOR:HORIZONTAL:BASIC:V1:STATE ON
:CURSOR:HORIZONTAL:BASIC:V1:STATE? ->
:CURSOR:HORIZONTAL:BASIC:V1:STATE 1
```

:CURSor:HORizontal[:BASic]:V<x>:VALue?

Function Queries the voltage value at the horizontal cursor.

Syntax
```
:CURSor:HORizontal[:BASic]:V<x>:VALue?
```
<x> = 1 or 2

Example
```
:CURSOR:HORIZONTAL:BASIC:V1:VALUE? ->
:CURSOR:HORIZONTAL:BASIC:V1:
VALUE 1.000E+00
```

:CURSor:HORizontal:CALCulation?

Function Queries all settings related to calculation items of the horizontal cursors.

Syntax
```
:CURSor:HORizontal:CALCulation?
```

Example
```
:CURSOR:HORIZONTAL:CALCULATION? ->
:CURSOR:HORIZONTAL:CALCULATION:
DEFINE1 “V1”;DEFINE2 “V2”;DEFINE3 “V1”;
DEFINE4 “V2”;STATE1 0;STATE2 0;
STATE3 0;STATE4 0
```

:CURSor:HORizontal:CALCulation:ALL

Function Turns ON/OFF all calculation items of the horizontal cursors.

Syntax
```
:CURSor:HORizontal:CALCulation:ALL
{<Boolean>}
```

Example
```
:CURSOR:HORIZONTAL:CALCULATION:ALL ON
```

:CURSor:HORizontal:CALCulation:DEFine<x>

Function Sets the equation of the calculation item of the horizontal cursor or queries the current setting.

Syntax
```
:CURSor:HORizontal:CALCulation:
DEFine<x> {<String>}
:CURSor:HORizontal:CALCulation:
DEFine<x>?
```
<x> = 1 to 4
<String> = Up to 128 characters

Example
```
:CURSOR:HORIZONTAL:CALCULATION:
DEFINE1 “V1”
:CURSOR:HORIZONTAL:CALCULATION:DEFINE1?
-> :CURSOR:HORIZONTAL:CALCULATION:
DEFINE1 “V1”
```

:CURSor:HORizontal:CALCulation:STATe<x>

Function Turns ON/OFF the calculation item of the horizontal cursor or queries the current setting.

Syntax
```
:CURSor:HORizontal:CALCulation:STATe<x>
{<Boolean>}
:CURSor:HORizontal:CALCulation:STATe<x>?
```
<x> = 1 to 4

Example
```
:CURSOR:HORIZONTAL:CALCULATION:STATE1 ON
:CURSOR:HORIZONTAL:CALCULATION:STATE1?
-> :CURSOR:HORIZONTAL:CALCULATION:
STATE1 1
```

:CURSor:HORizontal:CALCulation:VALue<x>?

Function Queries the measured value of the calculation item of the horizontal cursor.

Syntax
```
:CURSor:HORizontal:CALCulation:
VALue<x>?
```
<x> = 1 to 4

Example
```
:CURSOR:HORIZONTAL:CALCULATION:VALUE1?
-> :CURSOR:HORIZONTAL:CALCULATION:
VALUE1 0.000E+00
```

:CURSor:HORizontal:TRACe

Function Sets the source trace of the horizontal cursor or queries the current setting.

Syntax
```
:CURSor:HORizontal:TRACe {<NRf>}
:CURSor:HORizontal:TRACe?
```
<NRf> = 1 to 8

Example
```
:CURSOR:HORIZONTAL:TRACE 1
:CURSOR:HORIZONTAL:TRACE? ->
:CURSOR:HORIZONTAL:TRACE 1
```

:CURSor:MARKer?

Function Queries all settings related to the marker cursors.

Syntax
```
:CURSor:MARKer?
```

Example
```
:CURSOR:MARKER? -> :CURSOR:MARKER:
CALCULATION:DEFINE1 “T2”;DEFINE2 “V2”;
DEFINE3 “V3”;DEFINE4 “V4”;STATE1 0;
STATE2 0;STATE3 0;STATE4 0;:CURSOR:
MARKER:CM1:DT2:STATE 1;:CURSOR:MARKER:
CM1:DT3:STATE 1;:CURSOR:MARKER:CM1:DT4:
STATE 1;:CURSOR:MARKER:CM1:DV2:
STATE 1;:CURSOR:MARKER:CM1:DV3:
STATE 1;:CURSOR:MARKER:CM1:DV4:
STATE 1;:CURSOR:MARKER:CM1:
POSITION 1.000E+00;T:STATE 1;:CURSOR:
MARKER:CM1:TRACE 1;V:STATE 1;:CURSOR:
MARKER:CM2:DT1:STATE 1;:CURSOR:MARKER:
CM2:DT3:STATE 1;:CURSOR:MARKER:CM2:DT4:
STATE 1;:CURSOR:MARKER:CM2:DV1:
STATE 1;:CURSOR:MARKER:CM2:DV3:
STATE 1;:CURSOR:MARKER:CM2:DV4:
STATE 1;:CURSOR:MARKER:CM2:
POSITION 1.000E+00;T:STATE 1;:CURSOR:
MARKER:CM2:TRACE 1;V:STATE 1;:CURSOR:
MARKER:CM3:DT1:STATE 1;:CURSOR:MARKER:
CM3:DT2:STATE 1;:CURSOR:MARKER:CM3:DT4:
STATE 1;:CURSOR:MARKER:CM3:DV1:
STATE 1;:CURSOR:MARKER:CM3:DV2:
STATE ......
```

:CURSor:MARKer:CALCulation?

Function Queries all settings related to calculation items of the marker cursors.

Syntax
```
:CURSor:MARKer:CALCulation?
```

Example
```
:CURSOR:MARKER:CALCULATION? ->
:CURSOR:MARKER:CALCULATION:
DEFINE1 “T2”;DEFINE2 “V2”;DEFINE3 “V3”;
DEFINE4 “V4”;STATE1 0;STATE2 0;
STATE3 0;STATE4 0
```

:CURSor:MARKer:CALCulation:ALL

Function Turns ON/OFF all calculation items of the marker cursors.

Syntax
```
:CURSor:MARKer:CALCulation:ALL
{<Boolean>}
```

Example
```
:CURSOR:MARKER:CALCULATION:ALL ON
```

:CURSor:MARKer:CALCulation:DEFine<x>

Function Sets the equation of the calculation item of the marker cursors or queries the current setting.

Syntax
```
:CURSor:MARKer:CALCulation:DEFine<x>
{<String>}
:CURSor:MARKer:CALCulation:DEFine<x>?
```
<x> = 1 to 4
<String> = Up to 128 characters

Example
```
:CURSOR:MARKER:CALCULATION:DEFINE1 “T1”
:CURSOR:MARKER:CALCULATION:DEFINE1? ->
:CURSOR:MARKER:CALCULATION:DEFINE1 “T1”
```

:CURSor:MARKer:CALCulation:STATe<x>

Function Turns ON/OFF the calculation item of the marker cursors or queries the current setting.

Syntax
```
:CURSor:MARKer:CALCulation:STATe<x>
{<Boolean>}
:CURSor:MARKer:CALCulation:STATe<x>?
```
<x> = 1 to 4

Example
```
:CURSOR:MARKER:CALCULATION:STATE1 ON
:CURSOR:MARKER:CALCULATION:STATE1? ->
:CURSOR:MARKER:CALCULATION:STATE1 1
```

:CURSor:MARKer:CALCulation:VALue<x>?

Function Queries the measured value of the calculation item of the marker cursors.

Syntax
```
:CURSor:MARKer:CALCulation:VALue<x>?
```
<x> = 1 to 4

Example
```
:CURSOR:MARKER:CALCULATION:VALUE1? ->
:CURSOR:MARKER:CALCULATION:VALUE1
0.000E+00
```

:CURSor:MARKer:CM<x>?

Function Queries all settings related to the marker cursor.

Syntax
```
:CURSor:MARKer:CM<x>?
```
<x> = 1 to 4

Example
```
:CURSOR:MARKER:CM1? ->
:CURSOR:MARKER:CM1:DT2:STATE 1;:CURSOR:
MARKER:CM1:DT3:STATE 1;:CURSOR:MARKER:
CM1:DT4:STATE 1;:CURSOR:MARKER:CM1:DV2:
STATE 1;:CURSOR:MARKER:CM1:DV3:
STATE 1;:CURSOR:MARKER:CM1:DV4:
STATE 1;:CURSOR:MARKER:CM1:
POSITION 1.000E+00;T:STATE 1;:CURSOR:
MARKER:CM1:TRACE 1;V:STATE 1
```

:CURSor:MARKer:CM<x>:ALL

Function Turns ON/OFF all items of the marker cursor.
Syntax :CURSor:MARKer:CM<x>ALL {<Boolean>}
<x> = 1 to 4
Example :CURSOR:MARKER:CM1:ALL ON

:CURSor:MARKer:CM<x>:DT<x>?

Function Queries all settings related to the ΔT measurement of the cursor marker.
Syntax :CURSor:MARKer:CM<x>:DT<x>?
<x> of CM<x> = 1 to 4
<x> of DT<x> = 1 to 4
Example :CURSOR:MARKER:CM1:DT1? ->
:CURSOR:MARKER:CM1:DT1:STATE 1

:CURSor:MARKer:CM<x>:DT<x>:STATe

Function Turns ON/OFF the ΔT measurement of the maker cursor or queries the current setting.
Syntax :CURSor:MARKer:CM<x>:DT<x>:STATe {<Boolean>}
:CURSor:MARKer:CM<x>:DT<x>:STATe?
<x> of CM<x> = 1 to 4
<x> of DT<x> = 1 to 4
Example :CURSOR:MARKER:CM1:DT2:STATE ON
:CURSOR:MARKER:CM1:DT2:STATE? ->
:CURSOR:MARKER:CM1:DT2:STATE 1

:CURSor:MARKer:CM<x>:DT<x>:VALue?

Function Queries the ΔT value of the marker cursor.
Syntax :CURSor:MARKer:CM<x>:DT<x>:VALue?
<x> of CM<x> = 1 to 4
<x> of DT<x> = 1 to 4
Example :CURSOR:MARKER:CM1:DT2:VALUE? ->
:CURSOR:MARKER:CM1:DT2:VALUE 0.000E+00

:CURSor:MARKer:CM<x>:DV<x>?

Function Queries all settings related to the ΔV measurement of the cursor marker.
Syntax :CURSor:MARKer:CM<x>:DV<x>?
<x> of CM<x> = 1 to 4
<x> of DV<x> = 1 to 4
Example :CURSOR:MARKER:CM1:DV2? ->
:CURSOR:MARKER:CM1:DV2:STATE 1

:CURSor:MARKer:CM<x>:DV<x>:STATe

Function Turns ON/OFF the ΔV measurement of the maker cursor or queries the current setting.
Syntax :CURSor:MARKer:CM<x>:DV<x>:STATe {<Boolean>}
:CURSor:MARKer:CM<x>:DV<x>:STATe?
<x> of CM<x> = 1 to 4
<x> of DV<x> = 1 to 4
Example :CURSOR:MARKER:CM1:DV2:STATE ON
:CURSOR:MARKER:CM1:DV2:STATE? ->
:CURSOR:MARKER:CM1:DV2:STATE 1

:CURSor:MARKer:CM<x>:DV<x>:VALue?

Function Queries the ΔV value of the marker cursor.
Syntax :CURSor:MARKer:CM<x>:DV<x>:VALue?
<x> of CM<x> = 1 to 4
<x> of DV<x> = 1 to 4
Example :CURSOR:MARKER:CM1:DV2:VALUE? ->
:CURSOR:MARKER:CM1:DV2:VALUE 0

:CURSor:MARKer:CM<x>:JUMP

Function Jumps the marker cursor to the center position of the zoom waveform.
Syntax :CURSor:MARKer:CM<x>:JUMP {Z1|Z2}
<x> = 1 to 4
Example :CURSOR:MARKER:CM1:JUMP Z1

:CURSor:MARKer:CM<x>:POSition

Function Sets the marker cursor position or queries the current setting.
Syntax :CURSor:MARKer:CM<x>:POSition {<NRf>}
:CURSor:MARKer:CM<x>:POSition?
<x> = 1 to 4
<NRf> = –5 to 5 div
Example :CURSOR:MARKER:CM1:POSITION 1
:CURSOR:MARKER:CM1:POSITION? ->
:CURSOR:MARKER:CM1:POSITION 1.000E+00

:CURSor:MARKer:CM<x>:T?

Function Queries all settings related to the time measurement of the marker cursor.
Syntax :CURSor:MARKer:CM<x>:T?
<x> = 1 to 4
Example :CURSOR:MARKER:CM1:T? ->
:CURSOR:MARKER:CM1:T:STATE 1

:CURSor:MARKer:CM<x>:T:STATe

Function Turns ON/OFF the time measurement of the maker cursor or queries the current setting.

Syntax
```
:CURSor:MARKer:CM<x>:T:STATe {<Boolean>}
:CURSor:MARKer:CM<x>:T:STATe?
```
<x> = 1 to 4

Example
```
:CURSOR:MARKER:CM1:T:STATE ON
:CURSOR:MARKER:CM1:T:STATE? ->
:CURSOR:MARKER:CM1:T:STATE 1
```

:CURSor:MARKer:CM<x>:T:VALue?

Function Queries the time value at the marker cursor position.

Syntax
```
:CURSor:MARKer:CM<x>:T:VALue?
```
<x> = 1 to 4

Example
```
:CURSOR:MARKER:CM1:T:VALUE? ->
:CURSOR:MARKER:CM1:T:VALUE 0.000E+00
```

:CURSor:MARKer:CM<x>:TRACe

Function Sets the source trace of the marker cursor or queries the current setting.

Syntax
```
:CURSor:MARKer:CM<x>:TRACe {<NRf>|OFF}
:CURSor:MARKer:CM<x>:TRACe?
```
<x> = 1 to 4
<NRf> = 1 to 8

Example
```
:CURSOR:MARKER:CM1:TRACE 1
:CURSOR:MARKER:CM1:TRACE? ->
:CURSOR:MARKER:CM1:TRACE 1
```

:CURSor:MARKer:CM<x>:V?

Function Queries all settings related to the voltage measurement of the marker cursor.

Syntax
```
:CURSor:MARKer:CM<x>:V?
```
<x> = 1 to 4

Example
```
:CURSOR:MARKER:CM1:V? ->
:CURSOR:MARKER:CM1:V:STATE 1
```

:CURSor:MARKer:CM<x>:V:STATe

Function Turns ON/OFF the voltage measurement of the maker cursor or queries the current setting.

Syntax
```
:CURSor:MARKer:CM<x>:V:STATe {<Boolean>}
:CURSor:MARKer:CM<x>:V:STATe?
```
<x> = 1 to 4

Example
```
:CURSOR:MARKER:CM1:V:STATE ON
:CURSOR:MARKER:CM1:V:STATE? ->
:CURSor:MARKER:CM1:V:STATE 1
```

:CURSor:MARKer:CM<x>:V:VALue?

Function Queries the voltage value at the marker cursor position.

Syntax
```
:CURSor:MARKer:CM<x>:V:VALue?
```
<x> = 1 to 4

Example
```
:CURSOR:MARKER:CM1:V:VALUE? ->
:CURSOR:MARKER:CM1:V:VALUE 0.000E+00
```

:CURSor:SERial?

Function Queries all settings related to the serial cursors.

Syntax
```
:CURSor:SERial?
```

Example
```
:CURSOR:SERIAL? ->
:CURSOR:SERIAL:SCURSOR1:ACTIVE HIGH;
BCOUNT 8;BITRATE 10.00E+00;
BITORDER MSBFIRST;FORMAT BINARY;
HYSTERESIS 1.000E+00;LEVEL 0.000E+00;
MODE 1;POSITION -4.000E+00;TRACE 1;
TRACK OFF;:CURSOR:SERIAL:SCURSOR2:
ACTIVE HIGH;BCOUNT 8;BITRATE 1.000E+00;
BITORDER MSBFIRST;FORMAT BINARY;
HYSTERESIS 3.000E+00;LEVEL 0.000E+00;
MODE 0;POSITION -4.000E+00;TRACE 2;
TRACK OFF
```

:CURSor:SERial:SCURsor<x>?

Function Queries all settings related to the serial cursor.

Syntax
```
:CURSor:SERial:SCURsor<x>?
```
<x> = 1 or 2

Example
```
:CURSOR:SERIAL:SCURSOR1? ->
:CURSOR:SERIAL:SCURSOR1:ACTIVE HIGH;
BCOUNT 8;BITRATE 1.25E+06;
BITORDER MSBFIRST;FORMAT BINARY;
HYSTERESIS 300.00E-03;LEVEL 0.000E+00;
MODE 1;POSITION -4.00E+00;TRACE 1;
TRACK OFF
```

:CURSor:SERial:SCURsor<x>:ACTive

Function Sets the active level of the serial cursor or queries the current setting.

Syntax
```
:CURSor:SERial:SCURsor<x>:ACTive {HIGH|LOW}
:CURSor:SERial:SCURsor<x>:ACTive?
```
<x> = 1 or 2

Example
```
:CURSOR:SERIAL:SCURSOR1:ACTIVE HIGH
:CURSOR:SERIAL:SCURSOR1:ACTIVE? ->
:CURSOR:SERIAL:SCURSOR1:ACTIVE HIGH
```

:CURSor:SERial:SCURsor<x>:BCOunt

Function Sets the bit length of the serial cursor or queries the current setting.

Syntax
```
:CURSor:SERial:SCURsor<x>:BCOunt {<NRf>}
:CURSor:SERial:SCURsor<x>:BCOunt?
```
<x> = 1 or 2
<NRf> = 1 to 128 (bits)

Example
```
:CURSOR:SERIAL:SCURSOR1:BCOUNT 8
:CURSOR:SERIAL:SCURSOR1:BCOUNT? ->
:CURSOR:SERIAL:SCURSOR1:BCOUNT 8
```

:CURSor:SERial:SCURsor<x>:BITRate

Function Sets the bit rate of the serial cursor or queries the current setting.

Syntax
```
:CURSor:SERial:SCURsor<x>:BITRate
{<NRf>}
:CURSor:SERial:SCURsor<x>:BITRate?
```
<x> = 1 or 2
<NRf> = 1 to 1 G (bps)

Example
```
:CURSOR:SERIAL:SCURSOR1:BITRATE 10
:CURSOR:SERIAL:SCURSOR1:BITRATE? ->
:CURSOR:SERIAL:SCURSOR1:
BITRATE 10.00E+00
```

:CURSor:SERial:SCURsor<x>:BITorder

Function Sets the bit order of the serial cursor or queries the current setting.

Syntax
```
:CURSor:SERial:SCURsor<x>:BITorder
{LSBFirst|MSBFirst}
:CURSor:SERial:SCURsor<x>:BITorder?
```
<x> = 1 or 2

Example
```
:CURSOR:SERIAL:SCURSOR1:
BITORDER LSBFIRST
:CURSOR:SERIAL:SCURSOR1:BITORDER? ->
:CURSOR:SERIAL:SCURSOR1:
BITORDER LSBFIRST
```

:CURSor:SERial:SCURsor<x>:FORMat

Function Sets the display format of the serial cursor or queries the current setting.

Syntax
```
:CURSor:SERial:SCURsor<x>:FORMat
{BINary|HEXa}
:CURSor:SERial:SCURsor<x>:FORMat?
```
<x> = 1 or 2

Example
```
:CURSOR:SERIAL:SCURSOR1:FORMAT HEXA
:CURSOR:SERIAL:SCURSOR1:FORMAT? ->
:CURSOR:SERIAL:SCURSOR1:FORMAT HEXA
```

:CURSor:SERial:SCURsor<x>:HYSTeresis

Function Sets the hysteresis of the serial cursor or queries the current setting.

Syntax
```
:CURSor:SERial:SCURsor<x>:HYSTeresis
{<NRf>}
:CURSor:SERial:SCURsor<x>:HYSTeresis?
```
<x> = 1 or 2
<NRf> = 0 to 4 (div)

Example
```
:CURSOR:SERIAL:SCURSOR1:HYSTERESIS 1
:CURSOR:SERIAL:SCURSOR1:HYSTERESIS? ->
:CURSOR:SERIAL:SCURSOR1:
HYSTERESIS 1.000E+00
```

:CURSor:SERial:SCURsor<x>:JUMP

Function Moves the serial cursor to the specified direction.

Syntax
```
:CURSor:SERial:SCURsor<x>:JUMP
{BACK|FRONt}
```
<x> = 1 or 2

Example
```
:CURSOR:SERIAL:SCURSOR1:JUMP BACK
```

:CURSor:SERial:SCURsor<x>:LEVel

Function Sets the threshold level of the serial cursor or queries the current setting.

Syntax
```
:CURSor:SERial:SCURsor<x>:LEVel
{<Voltage>|<Current>}
:CURSor:SERial:SCURsor<x>:LEVel?
```
<x> = 1 or 2
<Voltage> and <Current> = See the DL9000 User's Manual.

Example
```
:CURSOR:SERIAL:SCURSOR1:LEVEL 0V
:CURSOR:SERIAL:SCURSOR1:LEVEL? ->
:CURSOR:SERIAL:SCURSOR1:LEVEL 0.000E+00
```

:CURSor:SERial:SCURsor<x>:MODE

Function Turns ON/OFF the serial cursor or queries the current setting.

Syntax
```
:CURSor:SERial:SCURsor<x>:MODE
{<Boolean>}
:CURSor:SERial:SCURsor<x>:MODE?
```
<x> = 1 or 2

Example
```
:CURSOR:SERIAL:SCURSOR1:MODE ON
:CURSOR:SERIAL:SCURSOR1:MODE? ->
:CURSOR:SERIAL:SCURSOR1:MODE 1
```

:CURSor:SERial:SCURsor<x>:POSition

Function Sets the serial cursor position or queries the current setting.

Syntax
```
:CURSor:SERial:SCURsor<x>:POSition
{<NRf>}
:CURSor:SERial:SCURsor<x>:POSition?
```
<x> = 1 or 2
<NRf> = –5 to 5 div

Example
```
:CURSOR:SERIAL:SCURSOR1:POSITION 1
:CURSOR:SERIAL:SCURSOR1:POSITION? ->
:CURSOR:SERIAL:SCURSOR1:
POSITION 1.000E+00
```

:CURSor:SERial:SCURsor<x>:TRACE

Function Sets the trace of the serial cursor or queries the current setting.

Syntax
```
:CURSor:SERial:SCURsor<x>:TRACE {<NRf>}
:CURSor:SERial:SCURsor<x>:TRACE?
```
<x> = 1 or 2
<NRf> = 1 to 8

Example
```
:CURSOR:SERIAL:SCURSOR1:TRACE 1
:CURSOR:SERIAL:SCURSOR1:TRACE? ->
:CURSOR:SERIAL:SCURSOR1:TRACE 1
```

:CURSor:SERial:SCURsor<x>:TRACK

Function Jumps the serial cursor onto the zoom waveform.

Syntax
```
:CURSor:SERial:SCURsor<x>:TRACK {OFF|Z1|Z2}
:CURSor:SERial:SCURsor<x>:TRACK?
```
<x> = 1 or 2

Example
```
:CURSOR:SERIAL:SCURSOR1:TRACK OFF
:CURSOR:SERIAL:SCURSOR1:TRACK? ->
:CURSOR:SERIAL:SCURSOR1:TRACK OFF
```

:CURSor:SERial:SCURsor<x>:VALue?

Function Queries the measured value of the serial cursor.

Syntax
```
:CURSor:SERial:SCURsor<x>:VALue?
```
<x> = 1 or 2

Example
```
:CURSOR:SERIAL:SCURSOR1:VALUE? ->
:CURSOR:SERIAL:SCURSOR1:VALUE 11000111
```

:CURSor:TYPE

Function Sets the cursor type or queries the current setting.

Syntax
```
:CURSor:TYPE {HORizontal|HAVertical|MARKer|SERial|VERTical|VT}
:CURSor:TYPE?
```

Example
```
:CURSOR:TYPE HORIZONTAL
:CURSOR:TYPE? ->
:CURSOR:TYPE HORIZONTAL
```

:CURSor:VERTical?

Function Queries all settings related to the vertical cursors.

Syntax
```
:CURSor:VERTical?
```

Example
```
:CURSOR:VERTICAL? -> :CURSOR:VERTICAL:
BASIC:DT:STATE 1;:CURSOR:VERTICAL:
BASIC:LINKAGE 1;PERDT:STATE 1;:CURSOR:
VERTICAL:BASIC:T1:POSITION -4.000E+00;
STATE 1;:CURSOR:VERTICAL:BASIC:T2:
POSITION -4.000E+00;STATE 1;:CURSOR:
VERTICAL:CALCULATION:DEFINE1 "T1";
DEFINE2 "T2";DEFINE3 "T1";DEFINE4 "T2";
STATE1 0;STATE2 0;STATE3 0;STATE4 0
```

:CURSor:VERTical:BASic?

Function Queries all settings related to basic items of the vertical cursors.

Syntax
```
:CURSor:VERTical:BASic?
```

Example
```
:CURSOR:VERTICAL:BASIC? ->
:CURSOR:VERTICAL:BASIC:DT:STATE 1;:
CURSOR:VERTICAL:BASIC:LINKAGE 1;PERDT:
STATE 1;:CURSOR:VERTICAL:BASIC:T1:
POSITION -4.000E+00;STATE 1;:CURSOR:
VERTICAL:BASIC:T2:POSITION -4.000E+00;
STATE 1
```

:CURSor:VERTical[:BASic]:ALL

Function Turns ON/OFF all basic items of the vertical cursors.

Syntax
```
:CURSor:VERTical[:BASic]:ALL {<Boolean>}
```

Example
```
:CURSOR:VERTICAL:BASIC:ALL ON
```

:CURSor:VERTical[:BASic]:DT?

Function Queries all settings related to the ΔT measurement of the vertical cursors.

Syntax
```
:CURSor:VERTical[:BASic]:DT?
```

Example
```
:CURSOR:VERTICAL:BASIC:DT? ->
:CURSOR:VERTICAL:BASIC:DT:STATE 1
```

:CURSor:VERTical[:BASic]:DT:STATe

Function Turns ON/OFF the ΔT measurement of the vertical cursors or queries the current setting.

Syntax
```
:CURSor:VERTical[:BASic]:DT:STATe {<Boolean>}
:CURSor:VERTical[:BASic]:DT:STATe?
```

Example
```
:CURSOR:VERTICAL:BASIC:DT:STATE ON
:CURSOR:VERTICAL:BASIC:DT:STATE? ->
:CURSOR:VERTICAL:BASIC:DT:STATE 1
```

:CURSor:VERTical[:BASic]:DT:VALue?

Function Queries the ΔT value of the vertical cursors.

Syntax
```
:CURSor:VERTical[:BASic]:DT:VALue?
```

Example
```
:CURSOR:VERTICAL:BASIC:DT:VALUE? ->
:CURSOR:VERTICAL:BASIC:DT:
VALUE 1.000E+00
```

:CURSor:VERTical[:BASic]:LINKage

Function Turns ON/OFF the vertical cursor link or queries the current setting.

Syntax
```
:CURSor:VERTical[:BASic]:LINKage {<Boolean>}
:CURSor:VERTical[:BASic]:LINKage?
```

Example
```
:CURSOR:VERTICAL:BASIC:LINKAGE ON
:CURSOR:VERTICAL:BASIC:LINKAGE? ->
:CURSOR:VERTICAL:BASIC:LINKAGE 1
```

:CURSor:VERTical[:BASic]:PERDt?

Function Queries all settings related to the 1/ΔT measurement of the vertical cursors.

Syntax `:CURSor:VERTical[:BASic]:PERDt?`

Example
```
:CURSOR:VERTICAL:BASIC:PERDT? ->
:CURSOR:VERTICAL:BASIC:PERDT:STATE 1
```

:CURSor:VERTical[:BASic]:PERDt:STATe

Function Turns ON/OFF the 1/ΔT measurement of the vertical cursors or queries the current setting.

Syntax
```
:CURSor:VERTical[:BASic]:PERDt:STATe
{<Boolean>}
:CURSor:VERTical[:BASic]:PERDt:STATe?
```

Example
```
:CURSOR:VERTICAL:BASIC:PERDT:STATE ON
:CURSOR:VERTICAL:BASIC:PERDT:STATE? ->
:CURSOR:VERTICAL:BASIC:PERDT:STATE 1
```

:CURSor:VERTical[:BASic]:PERDt:VALue?

Function Queries the 1/ΔT value of the vertical cursors.

Syntax `CURSor:VERTical[:BASic]:PERDt:VALue?`

Example
```
CURSOR:VERTICAL:BASIC:PERDT:VALUE? ->
:CURSOR:VERTICAL:BASIC:PERDT:
VALUE 0.000E+00
```

:CURSor:VERTical[:BASic]:T<x>?

Function Queries all settings related to the vertical cursor.

Syntax
```
:CURSor:VERTical[:BASic]:T<x>?
```
<x> = 1 or 2

Example
```
:CURSOR:VERTICAL:BASIC:T1? ->
:CURSOR:VERTICAL:BASIC:T1:
POSITION -4.000E+00;STATE 1
```

:CURSor:VERTical[:BASic]:T<x>:JUMP

Function Jumps to the center position of the zoom waveform of the vertical cursor.

Syntax
```
:CURSor:VERTical[:BASic]:T<x>:JUMP {Z1|
Z2}
```
<x> = 1 or 2

Example `:CURSOR:VERTICAL:BASIC:T1:JUMP Z1`

:CURSor:VERTical[:BASic]:T<x>:POSition

Function Sets the vertical cursor position or queries the current setting.

Syntax
```
:CURSor:VERTical[:BASic]:T<x>:POSition
{<NRf>}
:CURSor:VERTical[:BASic]:T<x>:POSition?
```
<x> = 1 or 2
<NRf> = –5 to 5 (div)

Example
```
:CURSOR:VERTICAL:BASIC:T1:POSITION 1
:CURSOR:VERTICAL:BASIC:T1:POSITION? ->
:CURSOR:VERTICAL:BASIC:T1:
POSITION 1.000E+00
```

:CURSor:VERTical[:BASic]:T<x>:STATe

Function Turns ON/OFF the vertical cursor or queries the current setting.

Syntax
```
:CURSor:VERTical[:BASic]:T<x>:STATe
{<Boolean>}
:CURSor:VERTical[:BASic]:T<x>:STATe?
```
<x> = 1 or 2

Example
```
:CURSOR:VERTICAL:BASIC:T1:STATE ON
:CURSOR:VERTICAL:BASIC:T1:STATE? ->
:CURSOR:VERTICAL:BASIC:T1:STATE 1
```

:CURSor:VERTical[:BASic]:T<x>:VALue?

Function Queries the time value at the vertical cursor.

Syntax
```
:CURSor:VERTical[:BASic]:T<x>:VALue?
```
<x> = 1 or 2

Example
```
:CURSOR:VERTICAL:BASIC:T1:VALUE? ->
:CURSOR:VERTICAL:BASIC:T1:
VALUE 0.000E+00
```

:CURSor:VERTical:CALCulation?

Function Queries all settings related to calculation items of the vertical cursors.

Syntax `:CURSor:VERTical:CALCulation?`

Example
```
:CURSOR:VERTICAL:CALCULATION? ->
:CURSOR:VERTICAL:CALCULATION:
DEFINE1 “T1”;DEFINE2 “T2”;DEFINE3 “T1”;
DEFINE4 “T2”;STATE1 0;STATE2 0;
STATE3 0;STATE4 0
```

:CURSor:VERTical:CALCulation:ALL

Function Turns ON/OFF all calculation items of the vertical cursors.

Syntax
```
:CURSor:VERTical:CALCulation:ALL
{<Boolean>}
```

Example `:CURSOR:VERTICAL:CALCULATION:ALL ON`

:CURSor:VERTical:CALCulation:DEFine<x>

Function Sets the equation of the calculation item of the vertical cursor or queries the current setting.

Syntax
```
:CURSor:VERTical:CALCulation:DEFine<x>
{<String>}
:CURSor:VERTical:CALCulation:DEFine<x>?
```
<x> = 1 to 4
<String> = Up to 128 characters

Example
```
:CURSOR:VERTICAL:CALCULATION:
DEFINE1 "T1"
:CURSOR:VERTICAL:CALCULATION:DEFINE1?
-> :CURSOR:VERTICAL:CALCULATION:
DEFINE1 "T1"
```

:CURSor:VERTical:CALCulation:STATe<x>

Function Turns ON/OFF the calculation item of the vertical cursor or queries the current setting.

Syntax
```
:CURSor:VERTical:CALCulation:STATe<x>
{<Boolean>}
:CURSor:VERTical:CALCulation:STATe<x>?
```
<x> = 1 to 4

Example
```
:CURSOR:VERTICAL:CALCULATION:STATE1 ON
:CURSOR:VERTICAL:CALCULATION:STATE1? ->
:CURSOR:VERTICAL:CALCULATION:STATE1 1
```

:CURSor:VERTical:CALCulation:VALue<x>?

Function Queries the measured value of the calculation item of the vertical cursor.

Syntax
```
:CURSor:VERTical:CALCulation:VALue<x>?
```
<x> = 1 to 4

Example
```
:CURSOR:VERTICAL:CALCULATION:VALUE1? ->
:CURSOR:VERTICAL:CALCULATION:VALUE1
0.000E+00
```

:CURSor:VT?

Function Queries all settings related to the VT cursor.

Syntax
```
:CURSor:VT?
```

Example
```
:CURSOR:VT? ->
:CURSOR:VT:BASIC:POSITION 0.000E+00;T:
STATE 1;:CURSOR:VT:BASIC:V1:STATE 1;:
CURSOR:VT:BASIC:V2:STATE 1;:CURSOR:VT:
BASIC:V3:STATE 1;:CURSOR:VT:BASIC:V4:
STATE 1;:CURSOR:VT:BASIC:V5:STATE 1;:
CURSOR:VT:BASIC:V6:STATE 1;:CURSOR:VT:
BASIC:V7:STATE 1;:CURSOR:VT:BASIC:V8:
STATE 1;:CURSOR:VT:CALCULATION:
DEFINE1 "T1";DEFINE2 "V(C1)";
DEFINE3 "V(C2)";DEFINE4 "V(C3)";
STATE1 0;STATE2 0;STATE3 0;STATE4 0
```

:CURSor:VT[:BASic]?

Function Queries all settings related to basic items of the VT cursor.

Syntax
```
:CURSor:VT[:BASic]?
```

Example
```
:CURSOR:VT:BASIC? ->
:CURSOR:VT:BASIC:POSITION 0.000E+00;T:
STATE 1;:CURSOR:VT:BASIC:V1:STATE 1;:
CURSOR:VT:BASIC:V2:STATE 1;:CURSOR:VT:
BASIC:V3:STATE 1;:CURSOR:VT:BASIC:V4:
STATE 1;:CURSOR:VT:BASIC:V5:STATE 1;:
CURSOR:VT:BASIC:V6:STATE 1;:CURSOR:VT:
BASIC:V7:STATE 1;:CURSOR:VT:BASIC:V8:
STATE 1
```

:CURSor:VT[:BASic]:ALL

Function Turns ON/OFF all basic items of the VT cursor.

Syntax
```
:CURSor:VT[:BASic]:ALL {<Boolean>}
```

Example
```
:CURSOR:VT:BASIC:ALL ON
```

:CURSor:VT[:BASic]:POSition

Function Sets the VT cursor position or queries the current setting.

Syntax
```
:CURSor:VT[:BASic]:POSition {<NRf>}
:CURSor:VT[:BASic]:POSition?
```
<NRf> = –5 to 5 (div)

Example
```
:CURSOR:VT:BASIC:POSITION 1
:CURSOR:VT:BASIC:POSITION? ->
:CURSOR:VT:BASIC:POSITION 1.000E+00
```

:CURSor:VT[:BASic]:T?

Function Queries all settings related to the time value of the VT cursor.

Syntax
```
:CURSor:VT[:BASic]:T?
```

Example
```
:CURSOR:VT:BASIC:T? ->
:CURSOR:VT:BASIC:T:STATE 1
```

:CURSor:VT[:BASic]:T:STATe

Function Turns ON/OFF the time value of the VT cursor or queries the current setting.

Syntax
```
:CURSor:VT[:BASic]:T:STATe {<Boolean>}
:CURSor:VT[:BASic]:T:STATe?
```

Example
```
:CURSOR:VT:BASIC:T:STATE ON
:CURSOR:VT:BASIC:T:STATE? ->
:CURSOR:VT:BASIC:T:STATE 1
```

:CURSor:VT[:BASic]:T:VALue?

Function Queries the time value at the VT cursor.

Syntax
```
:CURSor:VT[:BASic]:T:VALue?
```

Example
```
:CURSOR:VT:BASIC:T:VALUE? ->
:CURSOR:VT:BASIC:T:VALUE 0.000E+00
```

:CURSor:VT[:BASic]:V<x>?

Function Queries all settings related to the voltage value of the VT cursor.

Syntax
```
:CURSor:VT[:BASic]:V<x>?
```
<x> = 1 to 8

Example
```
:CURSOR:VT:BASIC:V1? ->
:CURSOR:VT:BASIC:V1:STATE 1
```

:CURSor:VT[:BASic]:V<x>:STATe

Function Turns ON/OFF the voltage value of the VT cursor or queries the current setting.

Syntax
```
:CURSor:VT[:BASic]:V<x>:STATe
{<Boolean>}
:CURSor:VT[:BASic]:V<x>:STATe?
```
<x> = 1 to 8

Example
```
:CURSOR:VT:BASIC:V1:STATE ON
:CURSOR:VT:BASIC:V1:STATE? ->
:CURSOR:VT:BASIC:V1:STATE 1
```

:CURSor:VT[:BASic]:V<x>:VALue?

Function Queries the voltage value at the VT cursor.

Syntax
```
:CURSor:VT[:BASic]:V<x>:VALue?
```
<x> = 1 to 8

Example
```
:CURSOR:VT:BASIC:V1:VALUE? ->
:CURSOR:VT:BASIC:V1:VALUE 0.000E+00
```

:CURSor:VT:CALCulation?

Function Queries all settings related to calculation items of the VT cursor.

Syntax
```
:CURSor:VT:CALCulation?
```

Example
```
:CURSOR:VT:CALCULATION? ->
:CURSOR:VT:CALCULATION:DEFINE1 “T1”;
DEFINE2 “V(C1)”;DEFINE3 “V(C2)”;
DEFINE4 “V(C3)”;STATE1 0;STATE2 0;
STATE3 0;STATE4 0
```

:CURSor:VT:CALCulation:ALL

Function Turns ON/OFF all calculation items of the VT cursor.

Syntax
```
:CURSor:VT:CALCulation:ALL {<Boolean>}
```

Example
```
:CURSOR:VT:CALCULATION:ALL ON
```

:CURSor:VT:CALCulation:DEFine<x>

Function Sets the equation of the calculation item of the VT cursor or queries the current setting.

Syntax
```
:CURSor:VT:CALCulation:DEFine<x>
{<String>}
:CURSor:VT:CALCulation:DEFine<x>?
```
<x> = 1 to 4
<String> = Up to 128 characters

Example
```
:CURSOR:VT:CALCULATION:DEFINE1 “T1”
:CURSOR:VT:CALCULATION:DEFINE1? ->
:CURSOR:VT:CALCULATION:DEFINE1 “T1”
```

:CURSor:VT:CALCulation:STATe<x>

Function Turns ON/OFF the calculation item of the VT cursor or queries the current setting.

Syntax
```
:CURSor:VT:CALCulation:STATe<x>
{<Boolean>}
:CURSor:VT:CALCulation:STATe<x>?
```
<x> = 1 to 4

Example
```
:CURSOR:VT:CALCULATION:STATE1 ON
:CURSOR:VT:CALCULATION:STATE1? ->
:CURSOR:VT:CALCULATION:STATE1 0
```

:CURSor:VT:CALCulation:VALue<x>?

Function Queries the measured value of the calculation item of the VT cursor.

Syntax
```
:CURSor:VT:CALCulation:VALue<x>?
```
<x> = 1 to 4

Example
```
:CURSOR:VT:CALCULATION:VALUE1? ->
:CURSOR:VT:CALCULATION:VALUE1 0.000E+00
```

:CURSor:VT:JUMP

Function Jumps the VT cursor to the center position of the zoom waveform.

Syntax
```
:CURSor:VT:JUMP {Z1|Z2}
```

Example
```
:CURSOR:VT:JUMP Z1
```

5.10 DISPlay Group

:DISPlay?

Function Queries all settings related to the display.

Syntax :DISPlay?

Example :DISPLAY? -> :DISPLAY:ACCUMULATE:
GRADE INTENSITY;MODE 0;PERSISTENCE:
COUNT INFINITY;MODE COUNT;
TIME 1.000E+00;:DISPLAY:BLIGHT:
AUTOOFF 0;BRIGHTNESS 1;LCD 1;TIMEOUT 1;
:DISPLAY:COLOR:CHANNEL1 BLUE;
CHANNEL2 BGREEN;CHANNEL3 CYAN;
CHANNEL4 DBLUE;MATH1 GRAY;MATH2 GREEN;
MATH3 LBLUE;MATH4 LGREEN;
REFERENCE1 MAGENTA;REFERENCE2 MGREEN;
REFERENCE3 ORANGE;REFERENCE4 PINK;
:DISPLAY:FORMAT SINGLE;GRATICULE GRID;
INTENSITY:WAVEFORM 10;CURSOR 10;
GRID 10;MARKER 10;ZBOX 10;
:DISPLAY:INTERPOLATE SINE;MAPPING:
MODE AUTO;TRACE1 1;TRACE2 1;TRACE3 1;
TRACE4 1;TRACE5 1;TRACE6 1;TRACE7 1;
TRACE8 1

:DISPlay:ACCumulate?

Function Queries all settings related to the accumulated display of waveforms.

Syntax :DISPlay:ACCumulate?

Example :DISPLAY:ACCUMULATE? ->
:DISPLAY:ACCUMULATE:GRADE INTENSITY;
MODE 0;PERSISTENCE:COUNT INFINITY;
MODE COUNT;TIME 1.000E+00

:DISPlay:ACCumulate:GRADe

Function Sets the accumulate mode or queries the current setting.

Syntax :DISPlay:ACCumulate:GRADe {COLor|
INTensity}
:DISPlay:ACCumulate:GRADe?

Example :DISPLAY:ACCUMULATE:GRADE INTENSITY
:DISPLAY:ACCUMULATE:GRADE? ->
:DISPLAY:ACCUMULATE:GRADE INTENSITY

:DISPlay:ACCumulate:MODE

Function Turns ON/OFF the accumulate mode or queries the current setting.

Syntax :DISPlay:ACCumulate:MODE {<Boolean>}
:DISPlay:ACCumulate:MODE?

Example :DISPLAY:ACCUMULATE:MODE ON
:DISPLAY:ACCUMULATE:MODE? ->
:DISPLAY:ACCUMULATE:MODE 1

:DISPlay:ACCumulate:PERSistence?

Function Queries all settings related to persistence.

Syntax :DISPlay:ACCumulate:PERSistence?

Example :DISPLAY:ACCUMULATE:PERSISTENCE? ->
:DISPLAY:ACCUMULATE:PERSISTENCE:
COUNT INFINITY;MODE COUNT;
TIME 1.000E+00

:DISPlay:ACCumulate:PERSistence:COUNt

Function Sets the persistence count or queries the current setting.

Syntax :DISPlay:ACCumulate:PERSistence:COUNt
{<NRf>|INFinity}
:DISPlay:ACCumulate:PERSistence:COUNt?
<NRf> = 1 to (the maximum number of history pages at the current record length)

Example :DISPLAY:ACCUMULATE:PERSISTENCE:
COUNT INFINITY
:DISPLAY:ACCUMULATE:PERSISTENCE:COUNT?
-> :DISPLAY:ACCUMULATE:PERSISTENCE:
COUNT INFINITY

:DISPlay:ACCumulate:PERSistence:MODE

Function Sets the persistence mode or queries the current setting.

Syntax :DISPlay:ACCumulate:PERSistence:MODE
{COUNt|TIME}
:DISPlay:ACCumulate:PERSistence:MODE?

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

:DISPlay:ACCumulate:PERSistence:TIME

Function Sets the persistence time or queries the current setting.

Syntax :DISPlay:ACCumulate:PERSistence:TIME
{<Time>|INFinity}
:DISPlay:ACCumulate:PERSistence:TIME?
<Time> = 100 ms to 1 s (100 ms steps), 1 s to 10 s (200 ms steps), 10 s to 100 s (2 s steps)

Example :DISPLAY:ACCUMULATE:PERSISTENCE:TIME 1S
:DISPLAY:ACCUMULATE:PERSISTENCE:TIME?
-> :DISPLAY:ACCUMULATE:PERSISTENCE:
TIME 1.000E+00

:DISPlay:BLIGht?

Function Queries all settings related to the backlight.

```
Syntax  :DISPlay:BLIGht?
Example :DISPLAY:BLIGHT? ->
        :DISPLAY:BLIGHT:AUTOOFF 0;BRIGHTNESS 1;
        LCD 1;TIMEOUT 1
```

:DISPlay:BLIGht:AUTooff

Function Sets the function that automatically turns the backlight off or queries the current setting.

```
Syntax  :DISPlay:BLIGht:AUTooff {<Boolean>}
        :DISPlay:BLIGht:AUTooff?
Example :DISPLAY:BLIGHT:AUTOOFF ON
        :DISPLAY:BLIGHT:AUTOOFF? ->
        :DISPLAY:BLIGHT:AUTOOFF 1
```

:DISPlay:BLIGht:BRIGhtness

Function Sets the brightness of the backlight or queries the current setting.

```
Syntax  :DISPlay:BLIGht:BRIGhtness {<NRf>}
        :DISPlay:BLIGht:BRIGhtness?
        <NRf> = 1 to 8
Example :DISPLAY:BLIGHT:BRIGHTNESS 1
        :DISPLAY:BLIGHT:BRIGHTNESS? ->
        :DISPLAY:BLIGHT:BRIGHTNESS 1
```

:DISPlay:BLIGht:LCD

Function Turns ON/OFF the backlight or queries the current setting.

```
Syntax  :DISPlay:BLIGht:LCD {<Boolean>}
        :DISPlay:BLIGht:LCD?
Example :DISPLAY:BLIGHT:LCD ON
        :DISPLAY:BLIGHT:LCD? ->
        :DISPLAY:BLIGHT:LCD 1
```

:DISPlay:BLIGht:TIMeout

Function Sets the timeout of the backlight or queries the current setting.

```
Syntax  :DISPlay:BLIGht:TIMeout {<NRf>}
        :DISPlay:BLIGht:TIMeout?
        <NRf> = 1 to 60 (minutes)
Example :DISPLAY:BLIGHT:TIMEOUT 60
        :DISPLAY:BLIGHT:TIMEOUT? ->
        :DISPLAY:BLIGHT:TIMEOUT 60
```

:DISPlay:COLor?

Function Queries all settings related to the waveform display color.

```
Syntax  :DISPlay:COLor?
Example :DISPLAY:COLOR? ->
        :DISPLAY:COLOR:CHANNEL1 BLUE;
        CHANNEL2 BGREEN;CHANNEL3 CYAN;
        CHANNEL4 DBLUE;MATH1 GRAY;MATH2 GREEN;
        MATH3 LBLUE;MATH4 LGREEN;
        REFERENCE1 MAGENTA;REFERENCE2 MGREEN;
        REFERENCE3 ORANGE;REFERENCE4 PINK
```

:DISPlay:COLor:{CHANnel<x>|MATH<x>|REFerence<x>}

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

```
Syntax  :DISPlay:COLor:{CHANnel<x>|MATH<x>|
        REFerence<x>}
        {BLUE|BGReen|CYAN|DBLue|GRAY|GREen|
        LBLue|LGReen|MAGenta|MGReen|ORANge|
        PINK|PURPle|RED|SPINk|YELLow}
        :DISPlay:COLor:{CHANnel<x>|MATH<x>|
        REFerence<x>}?
        <x> of CHANnel<x> = 1 to 4
        <x> of MATH<x> = 1 to 4
        <x> of REFerence<x> = 1 to 4
Example :DISPLAY:COLOR:CHANNEL1 BLUE
        :DISPLAY:COLOR:CHANNEL1? ->
        :DISPLAY:COLOR:CHANNEL1 BLUE
```

:DISPlay:FORMat

Function Sets the display format or queries the current setting.

```
Syntax  :DISPlay:FORMat {DUAL|QUAD|SINGle|
        TRIad}
        :DISPlay:FORMat?
Example :DISPLAY:FORMAT SINGLE
        :DISPLAY:FORMAT? ->
        :DISPLAY:FORMAT SINGLE
```

:DISPlay:GRATicule

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

```
Syntax  :DISPlay:GRATicule {CROSshair|FRAMe|
        GRID|LINE}
        :DISPlay:GRATicule?
Example :DISPLAY:GRATICULE CROSSHAIR
        :DISPLAY:GRATICULE? ->
        :DISPLAY:GRATICULE CROSSHAIR
```

:DISPlay:INTENsity?

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

```
Syntax  :DISPlay:INTENsity?
        :DISPlay:INTENsity? ->
Example :DISPLAY:INTENSITY? ->
        :DISPLAY:INTENSITY:WAVEFORM 10;
        CURSOR 10;GRID 10;MARKER 10
```

:DISPlay:INTENsity[:WAVeform]

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

```
Syntax  :DISPlay:INTENsity[:WAVeform] {<NRf>}
        :DISPlay:INTENsity[:WAVeform]?
        <NRf> = 1 to 20
Example :DISPLAY:INTENSITY:WAVEFORM 10
        :DISPLAY:INTENSITY:WAVEFORM? ->
        :DISPLAY:INTENSITY:WAVEFORM 10
```

:DISPlay:INTENsity:{CURSor|GRID|MARKer|ZBOX}

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

```
Syntax  :DISPlay:INTENsity:{CURSor|GRID|
        MARKer|ZBOX} {<NRf>}
        :DISPlay:INTENsity:{CURSor|GRID|
        MARKer|ZBOX}?
        <NRf> = 0 to 31
Example :DISPLAY:INTENSITY:CURSOR 10
        :DISPLAY:INTENSITY:CURSOR? ->
        :DISPLAY:INTENSITY:CURSOR 10
```

:DISPlay:INTERpolate

Function Sets the display interpolation format or queries the current setting.

```
Syntax  :DISPlay:INTERpolate {LINE|OFF|PULSe|
        SINE}
        :DISPlay:INTERpolate?
Example :DISPLAY:INTERPOLATE SINE
        :DISPLAY:INTERPOLATE? ->
        :DISPLAY:INTERPOLATE SINE
```

:DISPlay:MAPPing?

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

```
Syntax  :DISPlay:MAPPing?
Example :DISPLAY:MAPPING? ->
        :DISPLAY:MAPPING:MODE AUTO;TRACE1 1;
        TRACE2 1;TRACE3 1;TRACE4 1;TRACE5 1;
        TRACE6 1;TRACE7 1;TRACE8 1
```

:DISPlay:MAPPing[:MODE]

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

```
Syntax  :DISPlay:MAPPing[:MODE] {AUTO|MANual}
        :DISPlay:MAPPing[:MODE]?
Example :DISPLAY:MAPPING:MODE AUTO
        :DISPLAY:MAPPING:MODE? ->
        :DISPLAY:MAPPING:MODE AUTO
```

:DISPlay:MAPPing:TRACe<x>

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

```
Syntax  :DISPlay:MAPPing:TRACe<x> {<NRf>}
        :DISPlay:MAPPing:TRACe<x>?
        <x> = 1 to 8
        <NRf> = 1 to 4
Example :DISPLAY:MAPPING:TRACE1 3
        :DISPLAY:MAPPING:TRACE1? ->
        :DISPLAY:MAPPING:TRACE1 3
```

5.11 FILEGroup

:FILE?

Function Queries all settings related to the specified storage medium or internal memory.

Syntax :FILE?

Example :FILE? -> :FILE:DIRECTORY:
CDIRECTORY "\Flash Mem\HOME";:FILE:
INTERNAL:STORE:BINARY:TRACE 1;:FILE:
INTERNAL:STORE:DMEMORY:TRACE 1;:FILE:
LOAD:BINARY:REFERENCE 1;:FILE:LOAD:
ZPOLYGON:ZONE 1;:FILE:LOAD:ZWAVE:
ZONE 1;:FILE:SAVE:AHISTOGRAM:
ANALYSIS 1;:FILE:SAVE:ANAMING DATE;
ASCII:COMPRESSION DECIMATION;
HISTORY ONE;TRACE 1;:FILE:SAVE:BINARY:
COMPRESSION DECIMATION;HISTORY ONE;
TRACE 1;:FILE:SAVE:
COMMENT "THIS IS TEST";DMEMORY:
TRACE 1;:FILE:SAVE:FFT:ANALYSIS 1;:
FILE:SAVE:FLOAT:COMPRESSION DECIMATION;
HISTORY ONE;TRACE 1;:FILE:SAVE:
NAME "DL9000";SBUS:ANALYSIS 1;:FILE:
SAVE:ZWAVE:ZONE 1

:FILE:DIRectory?

Function Queries all settings related to the specified storage medium.

Syntax :FILE:DIRectory?

Example :FILE:DIRECTORY? -> :FILE:DIRECTORY:
CDIRECTORY "\Flash Mem\HOME"

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

Function Sets the current directory or queries the current setting.

Syntax :FILE[:DIRectory]:CDIRectory {<String>}
:FILE[:DIRectory]:CDIRectory?
<String> = Up to 259 characters

Example :FILE:DIRECTORY:CDIRECTORY "\Flash Mem\DIR1\DIR2"
(Absolute path designation)
:FILE:DIRECTORY:CDIRECTORY "DIR2"
(Relative path designation)
:FILE:DIRECTORY:CDIRECTORY "\"
(Root directory designation)
:FILE:DIRECTORY:CDIRECTORY? ->
:FILE:DIRECTORY:
CDIRECTORY "\Flash Mem\DIR1\DIR2"

Description
- Data files cannot be saved to the root directory. Specify a save destination drive for the save directory.
  The following five drives are selectable.
  - Internal hard disk: "HD"
  - Internal memory: "Flash Mem"
  - Network drive: "Network"
  - PC card: "Storage Card<x>"
  - USB storage device: "USB Storage<x>"

  <x> = 1 to 4 (however, only a total of four "Storage Card" and "USB Storage" designations are supported (including partitions))
- Sets the specified directory the current directory for saving and loading.
- Absolute and relative path designations are possible.
- To specify an absolute path, enter a backslash at the front of the path.
- Relative path to higher level directories is not allowed.

:FILE[:DIRectory]:FREE?

Function Queries the free disk space in bytes on the specified storage medium.

Syntax :FILE[:DIRectory]:FREE?

Example :FILE:DIRECTORY:FREE?->
:FILE:DIRECTORY:FREE 65536

Description
- Returns the size of the current directory.
- If the current directory is the root directory, "0" is returned.

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

Function Creates a directory under the specified directory. This is an overlap command.

Syntax `:FILE[:DIRectory]:MDIRectory {<String>}`
<String> = Up to 259 characters

Example `:FILE:DIRECTORY:MDIRECTORY "\Flash Mem\DIR1\DIR2"`
(Absolute path designation)
`:FILE:DIRECTORY:MDIRECTORY "DIR2"`
(Relative path designation)

Description For details on the <String> parameter, see the description in `FILE[:DIRectory]:CDIRectory`.

:FILE:INTernal?

Function Queries all settings related to the internal memory.

Syntax `:FILE:INTernal?`

Example `:FILE:INTERNAL? ->`
`:FILE:INTERNAL:STORE:BINARY:TRACE 1;:FILE:INTERNAL:STORE:DMEMORY:TRACE 1`

:FILE:INTernal:RECall:{DMEMory|SETup}[:EXECute]

Function Executes the loading of specific data from the internal memory. This is an overlap command.

Syntax `:FILE:INTernal:RECall:{DMEMory|SETup}[:EXECute] {<NRf>}`
<NRf> = 1 to 4

Example (The following is an example for setup data.)
`:FILE:INTERNAL:RECALL:SETUP:EXECUTE 1`

:FILE:INTernal:STORe?

Function Queries all settings related to the saving to the internal memory.

Syntax `:FILE:INTernal:STORe?`

Example `:FILE:INTERNAL:STORE? ->`
`:FILE:INTERNAL:STORE:BINARY:TRACE 1;:FILE:INTERNAL:STORE:DMEMORY:TRACE 1`

:FILE:INTernal:STORe:{BINary|DMEMory}?

Function Queries all settings related to the saving of specific data to the internal memory.

Syntax `:FILE:INTernal:STORe:{BINary|DMEMory}?`

Example (The following is an example for binary data.)
`:FILE:INTERNAL:STORE:BINARY? ->`
`:FILE:INTERNAL:STORE:BINARY:TRACE 1`

:FILE:INTernal:STORe:{BINary|DMEMory|SETup}[:EXECute]

Function Executes the save operation of specific data to the internal memory. This is an overlap command.

Syntax `:FILE:INTernal:STORe:{BINary|DMEMory|SETup}[:EXECute] {<NRf>}`
<NRf> = 1 to 4

Example (The following is an example for binary data.)
`:FILE:INTERNAL:STORE:BINARY:EXECUTE 1`

:FILE:INTernal:STORe:BINary:TRACe

Function Sets the trace to be saved in binary data to the internal memory or queries the current setting.

Syntax `:FILE:INTernal:STORe:BINary:TRACe {<NRf>}`
`:FILE:INTernal:STORe:BINary:TRACe?`
<NRf> = 1 to 8

Example `:FILE:INTERNAL:STORE:BINARY:TRACE 1`
`:FILE:INTERNAL:STORE:BINARY:TRACE? ->`
`:FILE:INTERNAL:STORE:BINARY:TRACE 1`

:FILE:INTernal:STORe:DMEMory:TRACe

Function Sets the trace to be saved in accumulated data to the internal memory or queries the current setting.

Syntax `:FILE:INTernal:STORe:DMEMory:TRACe {<NRf>|ALL|XY1|XY2}`
`:FILE:INTernal:STORe:DMEMory:TRACe?`
<NRf> = 1 to 8

Example `:FILE:INTERNAL:STORE:DMEMORY:TRACE 1`
`:FILE:INTERNAL:STORE:DMEMORY:TRACE? ->`
`:FILE:INTERNAL:STORE:DMEMORY:TRACE 1`

:FILE:INTernal:UNLoad:DMEMory[:EXECute]

Function Clears the loaded accumulated data.

Syntax `:FILE:INTernal:UNLoad:DMEMory[:EXECute]`

Example `:FILE:INTERNAL:UNLOAD:DMEMORY:EXECUTE`

:FILE:LOAD?

Function Queries all settings related to the loading of files on the specified storage medium.

Syntax `:FILE:LOAD?`

Example `:FILE:LOAD? ->`
`:FILE:LOAD:BINARY:REFERENCE 1;:FILE:LOAD:ZPOLYGON:ZONE 1;:FILE:LOAD:ZWAVE:ZONE 1`

:FILE:LOAD:{BINary|ZPOLYGON|ZWAVe}?

Function Queries all settings related to the loading of specific data.

Syntax `:FILE:LOAD:{BINary|ZPOLYGON|ZWAVe}?`

Example (The following is an example for binary data.)
`:FILE:LOAD:BINARY? ->`
`:FILE:LOAD:BINARY:REFERENCE 1`

:FILE:LOAD:{BINary|DMEMory|MASK|SETup|ZPOLygon|ZWAVe}:ABORt

Function Aborts the loading of specific data.

Syntax
```
:FILE:LOAD:{BINary|DMEMory|MASK|SETup|
ZPOLygon|ZWAVe}:ABORt
```

Example (The following is an example for binary data.)
```
:FILE:LOAD:BINARY:ABORT
```

:FILE:LOAD:{BINary|DMEMory|MASK|SETup|ZPOLygon|ZWAVe}[:EXECute]

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

Syntax
```
:FILE:LOAD:{BINary|DMEMory|MASK|SETup|
ZPOLygon|ZWAVe}[:EXECute] {<String>}
```
<String> = Up to 259 characters

Example (The following is an example for binary data.)
```
:FILE:LOAD:BINARY:
EXECUTE "\Flash Mem\DIR\DATA"
```
(Absolute path designation)
```
:FILE:LOAD:BINARY:EXECUTE "DATA"
```
(Relative path designation)

Description For details on the <String> parameter, see the description in `FILE[:DIRectory]:CDIRectory`.

:FILE:LOAD:BINary:REFerence

Function Sets the load destination of the binary data or queries the current setting.

Syntax
```
:FILE:LOAD:BINary:REFerence {<NRf>|
ACQMemory}
:FILE:LOAD:BINary:REFerence?
```
<NRf> = 1 to 4

Example
```
:FILE:LOAD:BINARY:REFERENCE 1
:FILE:LOAD:BINARY:REFERENCE? ->
:FILE:LOAD:BINARY:REFERENCE 1
```

:FILE:LOAD:{ZPOLygon|ZWAVe}:ZONE

Function Sets the load destination of the zone data of specific data or queries the current setting.

Syntax
```
:FILE:LOAD:{ZPOLygon|ZWAVe}:
ZONE {<NRf>}
:FILE:LOAD:{ZPOLygon|ZWAVe}:ZONE?
```
<NRf> = 1 to 4

Example (The following is an example for ZONE WAVE data.)
```
:FILE:LOAD:ZWAVE:ZONE 1
:FILE:LOAD:ZWAVE:ZONE? ->
:FILE:LOAD:ZWAVE:ZONE 1
```

:FILE:SAVE?

Function Queries all settings related to the saving of data.

Syntax
```
:FILE:SAVE?
```

Example
```
:FILE:SAVE? -> :FILE:SAVE:AHISTOGRAM:
ANALYSIS 1;:FILE:SAVE:ANAMING DATE;
ASCII:COMPRESSION DECIMATION;
HISTORY ONE;TRACE 1;:FILE:SAVE:BINARY:
COMPRESSION DECIMATION;HISTORY ONE;
TRACE 1;:FILE:SAVE:
COMMENT "THIS IS TEST";DMEMORY:
TRACE 1;:FILE:SAVE:FFT:ANALYSIS 1;:
FILE:SAVE:FLOAT:COMPRESSION DECIMATION;
HISTORY ONE;TRACE 1;:FILE:SAVE:
NAME "DL9000";SBUS:ANALYSIS 1;:FILE:
SAVE:ZWAVE:ZONE 1
```

:FILE:SAVE:{AHIStogram|ASCii|BINary|DMEMory|FFT|FLOat|SBUS|ZWAVe}?

Function Queries all settings related to the saving of specific data.

Syntax
```
:FILE:SAVE:{AHIStogram|ASCii|BINary|
DMEMory|FFT|FLOat|SBUS|ZWAVe}?
```

Example (The following is an example for binary data.)
```
:FILE:SAVE:BINARY? ->
:FILE:SAVE:BINARY:COMPRESSION
DECIMATION;HISTORY ONE;TRACE 1
```

:FILE:SAVE:{AHIStogram|ASCii|BINary|DMEMory|FFT|FLOat|SETup|SBUS|WPARameter|ZWAVe}:ABORt

Function Aborts the save operation of specific data.

Syntax
```
:FILE:SAVE:{AHIStogram|ASCii|BINary|
DMEMory|FFT|FLOat|SETup|SBUS|
WPARameter|ZWAVe}:ABORt
```

Example (The following is an example for binary data.)
```
:FILE:SAVE:BINARY:ABORT
```

:FILE:SAVE:{AHIStogram|ASCii|BINary|DMEMory|FFT|FLOat|SETup|SBUS|WPARameter|ZWAVe}[:EXECute]

Function Executes the saving of specific data by specifying a file name. This is an overlap command.

Syntax
```
:FILE:SAVE:{AHIStogram|ASCii|BINary|
DMEMory|FFT|FLOat|SETup|SBUS|
WPARameter|ZWAVe}[:EXECute] {<String>}
```
<String> = Up to 259 characters

Example (The following is an example for binary data.)
```
:FILE:SAVE:BINARY:EXECUTE
:FILE:SAVE:BINARY:
EXECUTE "\Flash Mem\DIR\DATA"
```
(Absolute path designation)
```
:FILE:SAVE:BINARY:EXECUTE "DATA"
```
(Relative path designation)

Description
- If the path is not specified, the data is saved to the file name specified by :FILE:SAVE:NAME.
- If the path is specified, the auto naming function is disabled.
- For details on the <String> parameter, see the description in FILE[:DIRectory]:CDIRectory.

:FILE:SAVE:{AHIStogram|FFT|SBUS}:ANALysis

Function Sets the trace of specific data to be saved or queries the current setting.

Syntax
```
FILE:SAVE:{AHIStogram|FFT|SBUS}:
ANALysis {<NRf>}
FILE:SAVE:{AHIStogram|FFT|SBUS}:
ANALysis?
```
<NRf> = 1 to 8

Example (The following is an example for FFT data.)
```
:FILE:SAVE:FFT:ANALYSIS 1
:FILE:SAVE:FFT:ANALYSIS? ->
:FILE:SAVE:FFT:ANALYSIS 1
```

:FILE:SAVE:ANAMing

Function Sets the type of auto naming of file names of the saved data 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}:COMPression

Function Sets the compression method for saving specific data or queries the current setting.

Syntax
```
:FILE:SAVE:{ASCii|BINary|FLOat}:
COMPression {DECimation|OFF|PTOPeak}
:FILE:SAVE:{ASCii|BINary|FLOat}:
COMPression?
```

Example (The following is an example for binary data.)
```
:FILE:SAVE:BINARY:
COMPRESSION DECIMATION
:FILE:SAVE:BINARY:COMPRESSION? ->
:FILE:SAVE:BINARY:
COMPRESSION DECIMATION
```

:FILE:SAVE:{ASCii|BINary|FLOat}:HISTory

Function Sets whether to save the entire data or the selected data of specific data or queries the current setting.

Syntax
```
:FILE:SAVE:{ASCii|BINary|FLOat}:
HISTory {ALL|ONE}
:FILE:SAVE:{ASCii|BINary|FLOat}:
HISTory?
```

Example (The following is an example for binary data.)
```
:FILE:SAVE:BINARY:HISTORY ALL
:FILE:SAVE:BINARY:HISTORY? ->
:FILE:SAVE:BINARY:HISTORY ALL
```

:FILE:SAVE:{ASCii|BINary|FLOat}:TRACe

Function Sets the trace of specific data to be saved or queries the current setting.

Syntax
```
:FILE:SAVE:{ASCii|BINary|FLOat}:
TRACe {<NRf>|ALL}
:FILE:SAVE:{ASCii|BINary|FLOat}:TRACe?
```
<NRf> = 1 to 8

Example (The following is an example for binary data.)
```
:FILE:SAVE:BINARY:TRACE 1
:FILE:SAVE:BINARY:TRACE? ->
:FILE:SAVE:BINARY:TRACE 1
```

: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 160 characters

Example
```
:FILE:SAVE:COMMENT "THIS IS TEST"
:FILE:SAVE:COMMENT? ->
:FILE:SAVE:COMMENT "THIS IS TEST"
```

:FILE:SAVE:DMEMory:TRACe

Function Sets the trace to be saved as accumulated data or queries the current setting.

Syntax
```
:FILE:SAVE:DMEMory:TRACe {<NRf>|ALL|
XY1|XY2}
:FILE:SAVE:DMEMory:TRACe?
```
<NRf> = 1 to 8

Example
```
:FILE:SAVE:DMEMORY:TRACE 1
:FILE:SAVE:DMEMORY:TRACE? ->
:FILE:SAVE:DMEMORY:TRACE 1
```

:FILE:SAVE:NAME

Function Sets the name of the data file to be saved or queries the current setting.

Syntax
```
:FILE:SAVE:NAME {<Filename>}
:FILE:SAVE:NAME?
```

Example
```
:FILE:SAVE:NAME "DL9000"
:FILE:SAVE:NAME? ->
:FILE:SAVE:NAME "DL9000"
```

:FILE:SAVE:ZWAVe:ZONE

Function Sets the zone of the zone data to be saved or queries the current setting.

Syntax
```
:FILE:SAVE:ZWAVe:ZONE {<NRf>}
:FILE:SAVE:ZWAVe:ZONE?
```
<NRf> = 1 to 4

Example
```
:FILE:SAVE:ZWAVE:ZONE 1
:FILE:SAVE:ZWAVE:ZONE? ->
:FILE:SAVE:ZWAVE:ZONE 1
```

5.12 GONogo Group

:GONogo?

Function Queries all settings related to GO/NO-GO determination.

Syntax :GONogo?

Example :GONOGO? -> :GONOGO:ACTION:BUZZER 0;
HCOPY 0;SAVE 0;:GONOGO:
CONDITION1 DONTCARE;
CONDITION2 DONTCARE;
CONDITION3 DONTCARE;
CONDITION4 DONTCARE;LOGIC AND;MODE OFF;
SCONDITION:NGCOUNT 1;STOPCOUNT 1;:
GONOGO:TELECOMTEST:SELECT1:MASK:
ELEMENT1:WCOUNT 1.000E+00,2.000E+00;:
GONOGO:TELECOMTEST:SELECT2:MASK:
ELEMENT1:WCOUNT 1.000E+00,2.000E+00;:
GONOGO:TELECOMTEST:SELECT3:MASK:
ELEMENT1:WCOUNT 1.000E+00,2.000E+00;:
GONOGO:TELECOMTEST:SELECT4:MASK:
ELEMENT1:WCOUNT 2.000E+00,3.000E+00;:
GONOGO:ZPARAMETER:SELECT1:
MODE PARAMETER;PARAMETER:CATEGORY FFT;
FFT1:CALCULATION1 0.000E+00,1.000E+00;:
GONOGO:ZPARAMETER:SELECT1:PARAMETER:
MEASURE:STATISTICS MAXIMUM;:GONOGO:
ZPARAMETER:SELECT1:RECTANGLE:
HORIZONTAL 0.000E+00,1.000E+00;
VERTICAL 0.000E+00,1.000E+00;:GONOGO:
ZPARAMETER:SELECT1:TRACE 1;WAVE:
TRANGE 1.000E+00,2.000E+00;:GONOGO:
ZPARAMETER:SELECT1:WINDOW MAIN;:GONOGO:
ZPARAMETER:SELECT2:MODE PARAMETER;
PARAMETER:CATEGORY FFT;FFT1:
CALCULATION1 0.000E+00,1.000E+00...

:GONogo:ABORt

Function Aborts the GO/NO-GO determination.

Syntax :GONogo:ABORt

Example :GONOGO:ABORT

:GONogo:ACTion?

Function Queries all settings related to the action taken when the determination result is NO-GO and the criteria values.

Syntax :GONogo:ACTion?

Example :GONOGO:ACTION? ->
:GONOGO:ACTION:BUZZER 0;HCOPY 0;SAVE 0

:GONogo:ACTion:BUZZer

Function Sets whether to sound a buzzer when the determination result is NO-GO or queries the current setting.

Syntax :GONogo:ACTion:BUZZer {<Boolean>}
:GONogo:ACTion:BUZZer?

Example :GONOGO:ACTION:BUZZER ON
:GONOGO:ACTION:BUZZER? ->
:GONOGO:ACTION:BUZZER 1

:GONogo:ACTion:HCOPy

Function Sets whether to print the screen image on the printer when the determination result is NO-GO or queries the current setting.

Syntax :GONogo:ACTion:HCOPy {<Boolean>}
:GONogo:ACTion:HCOPy?

Example :GONOGO:ACTION:HCOPY ON
:GONOGO:ACTION:HCOPY? ->
:GONOGO:ACTION:HCOPY 1

:GONogo:ACTion:SAVE

Function Sets whether to save the waveform data to the storage medium when the determination result is NO-GO or queries the current setting.

Syntax :GONogo:ACTion:SAVE {<Boolean>}
:GONogo:ACTion:SAVE?

Example :GONOGO:ACTION:SAVE ON
:GONOGO:ACTION:SAVE? ->
:GONOGO:ACTION:SAVE 1

:GONogo:CONDition<x>

Function Sets the GO/NO-GO determination criteria or queries the current setting.

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

Example :GONOGO:CONDITION1 DONTCARE
:GONOGO:CONDITION1? ->
:GONOGO:CONDITION1 DONTCARE

:GONogo:COUNt?

Function Queries the actual number of GO/NO-GO determinations.

Syntax :GONogo:COUNt?

Example :GONOGO:COUNT? -> :GONOGO:COUNT 1

: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 GO/NO-GO determination logic 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 GO/NO-GO determination type or queries the current setting.

Syntax
```
GONogo:MODE {OFF|TELecomtest|
ZPARameter}
:GONogo:MODE?
```
Example
```
GONOGO:MODE OFF
:GONOGO:MODE? -> :GONOGO:MODE OFF
```

:GONogo:NGCount?

Function Queries the actual number of NO-GOs of the GO/NO-GO determination.

Syntax
```
:GONogo:NGCount?
```
Example
```
:GONOGO:NGCOUNT? -> :GONOGO:NGCOUNT 1
```

:GONogo[:SCONdition]? (Stop Condition)

Function Queries all settings related to the determination termination condition.

Syntax
```
:GONogo[:SCONdition]?
```
Example
```
:GONOGO:SCONDITION? ->
:GONOGO:SCONDITION:NGCOUNT 1;
STOPCOUNT 1
```

:GONogo[:SCONdition]:NGCount

Function Sets the number of NO-GOs that terminates the GO/NO-GO determination or queries the current setting.

Syntax
```
:GONogo[:SCONdition]:NGCount {<NRf>|
INFinite}
:GONogo[:SCONdition]:NGCount?
```
<NRf> = 1 to 1000

Example
```
:GONOGO:SCONDITION:NGCOUNT 1
:GONOGO:SCONDITION:NGCOUNT? ->
:GONOGO:SCONDITION:NGCOUNT 1
```

:GONogo[:SCONdition]:STOPcount

Function Sets the acquisition count that terminates the GO/NO-GO determination or queries the current setting.

Syntax
```
:GONogo[:SCONdition]:STOPcount {<NRf>|
INFinite}
:GONogo[:SCONdition]:STOPcount?
```
<NRf> = 1 to 1000000

Example
```
:GONOGO:SCONDITION:STOPCOUNT 1
:GONOGO:SCONDITION:STOPCOUNT? ->
:GONOGO:SCONDITION:STOPCOUNT 1
```

:GONogo:TELecomtest?

Function Queries all settings related to telecom test determination.

Syntax
```
:GONogo:TELecomtest?
```
Example
```
:GONOGO:TELECOMTEST? -> :GONOGO:
TELECOMTEST:SELECT1:MASK:ELEMENT1:
WCOUNT 2.000E+00,1.000E+00;:GONOGO:
TELECOMTEST:SELECT2:MASK:ELEMENT1:
WCOUNT2.000E+00,1.000E+00;:GONOGO:
TELECOMTEST:SELECT3:MASK:ELEMENT1:
WCOUNT 2.000E+00,1.000E+00;:GONOGO:
TELECOMTEST:SELECT4:MASK:ELEMENT1:
WCOUNT 2.000E+00,1.000E+00
```

:GONogo:TELecomtest:SELect<x>?

Function Queries all settings related to the condition of the telecom test determination.

Syntax
```
:GONogo:TELecomtest:SELect<x>?
```
<x> = 1 to 4

Example
```
:GONOGO:TELECOMTEST:SELECT1? ->
:GONOGO:TELECOMTEST:SELECT1:MASK:
ELEMENT1:PSPCOUNT 2.000E+00,1.000E+00
```

:GONogo:TELecomtest:SELect<x>: EYEPattern?

Function Queries all settings related to the eye pattern determination of the condition.

Syntax
```
:GONogo:TELecomtest:SELect<x>:
EYEPattern?
```
<x> = 1 to 4

Example
```
:GONOGO:TELECOMTEST:SELECT1:EYEPATTERN?
->:GONOGO:TELECOMTEST:SELECT1:
EYEPATTERN:DBERATE 2.000E+00,1.000E+00
```

:GONogo:TELecomtest:SELect<x>: EYEPattern:<Parameter>

Function Sets the upper and lower limits of the waveform parameter of the eye pattern determination or queries the current setting.

Syntax
```
:GONogo:TELecomtest:SELect<x>:
EYEPattern:<Parameter> {<NRf>,<NRf>|
<Voltage>,<Voltage>|
<Current>,<Current>|<Time>,<Time>}
:GONogo:TELecomtest:SELect<x>:
EYEPattern:<Parameter>?
```
<x> of SELect<x> = 1 to 4
```
<Parameter> =
{DBERate|EHEight|EWIDth|FALL|JITTer|
PCROssing|PDUTycycle|QFACtor|RISE|
SDBase|SDTop|T1CRossing|T2CRossing|
VBASe|VCRossing|VTOP}
```
<NRf>, <Voltage>, <Current>, and <Time> = See the DL9000 User's Manual (IM701310-01E).

Example
```
The following is an example for
DBERate.
:GONOGO:TELECOMTEST:SELECT1:EYEPATTERN:
DBERATE 1,2
:GONOGO:TELECOMTEST:SELECT1:EYEPATTERN:
DBERATE? ->
:GONOGO:TELECOMTEST:SELECT1:EYEPATTERN:
DBERATE 2.000E+00,1.000E+00
```

Description
- For the relationship between communication commands and parameters used by the DL9000, see appendix 4.
- For details on the parameters, see the DL9000 User's Manual (IM701310-01E).

:GONogo:TELecomtest:SELect<x>:MASK?

Function Queries all settings related to the mask determination of the condition.

Syntax
```
:GONogo:TELecomtest:SELect<x>:MASK?
```
<x> = 1 to 4

Example
```
:GONOGO:TELECOMTEST:SELECT1:MASK? ->
:GONOGO:TELECOMTEST:SELECT1:MASK:
ELEMENT1:PSPCOUNT 2.000E+00,1.000E+00
```

:GONogo:TELecomtest:SELect<x>:MASK: ELEMent<x>?

Function Queries all settings related to the element used in the mask determination.

Syntax
```
:GONogo:TELecomtest:SELect<x>:MASK:
ELEMent<x>?
```
<x> of SELect<x> = 1 to 4
<x> of ELEMent<x> = 1 to 4

Example
```
:GONOGO:TELECOMTEST:SELECT1:MASK:
ELEMENT1? -> :GONOGO:TELECOMTEST:
SELECT1:MASK:ELEMENT1:
PSPCOUNT 2.000E+00,1.000E+00
```

:GONogo:TELecomtest:SELect<x>:MASK: ELEMent<x>:PSPCount (Sample Point Count %)

Function Sets the upper and lower limits of the error rate for the number of sampled data points of the element or queries the current setting.

Syntax
```
:GONogo:TELecomtest:SELect<x>:MASK:
ELEMent<x>:PSPCount {<NRf>,<NRf>}
:GONogo:TELecomtest:SELect<x>:MASK:
ELEMent<x>:PSPCount?
```
<x> of SELect<x> = 1 to 4
<x> of ELEMent<x> = 1 to 4
<NRf> = 0 to 100 (%)

Example
```
:GONOGO:TELECOMTEST:SELECT1:MASK:
ELEMENT1:PSPCOUNT 1,2
:GONOGO:TELECOMTEST:SELECT1:MASK:
ELEMENT1:PSPCOUNT? ->
:GONOGO:TELECOMTEST:SELECT1:MASK:
ELEMENT1:PSPCOUNT 2.000E+00,1.000E+00
```

:GONogo:TELecomtest:SELect<x>:MASK: ELEMent<x>:PWCount (Wave Count %)

Function Sets the upper and lower limits of the error rate for the number of acquisitions of the element or queries the current setting.

Syntax
```
:GONogo:TELecomtest:SELect<x>:MASK:
ELEMent<x>:PWCount {<NRf>,<NRf>}
:GONogo:TELecomtest:SELect<x>:MASK:
ELEMent<x>:PWCount?
```
<x> of SELect<x> = 1 to 4
<x> of ELEMent<x> = 1 to 4
<NRf> = 0 to 100 (%)

Example
```
:GONOGO:TELECOMTEST:SELECT1:MASK:
ELEMENT1:PWCOUNT 1,2
:GONOGO:TELECOMTEST:SELECT1:MASK:
ELEMENT1:PWCOUNT? ->
:GONOGO:TELECOMTEST:SELECT1:MASK:
ELEMENT1:PWCOUNT 2.000E+00,1.000E+00
```

:GONogo:TELecomtest:SELect<x>:MASK:ELEMent<x>:SPCount (Sample Point Count)

Function Sets the upper and lower limits of the number of sampled data points for the element that results in error or queries the current setting.

Syntax
```
:GONogo:TELecomtest:SELect<x>:MASK:ELEMent<x>:SPCount {<NRf>,<NRf>}
:GONogo:TELecomtest:SELect<x>:MASK:ELEMent<x>:SPCount?
```
<x> of SELect<x> = 1 to 4
<x> of ELEMent<x> = 1 to 4
<NRf> = See the DL9000 User's Manual.

Example
```
:GONOGO:TELECOMTEST:SELECT1:MASK:ELEMENT1:SPCOUNT 1,2
:GONOGO:TELECOMTEST:SELECT1:MASK:ELEMENT1:SPCOUNT? ->
:GONOGO:TELECOMTEST:SELECT1:MASK:ELEMENT1:SPCOUNT 2.000E+00,1.000E+00
```

:GONogo:TELecomtest:SELect<x>:MASK:ELEMent<x>:WCOunt (Wave Count)

Function Sets the upper and lower limits of the number of acquisitions for the element that results in error or queries the current setting.

Syntax
```
:GONogo:TELecomtest:SELect<x>:MASK:ELEMent<x>:WCOunt {<NRf>,<NRf>}
:GONogo:TELecomtest:SELect<x>:MASK:ELEMent<x>:WCOunt?
```
<x> of SELect<x> = 1 to 4
<x> of ELEMent<x> = 1 to 4
<NRf> = See the DL9000 User's Manual.

Example
```
:GONOGO:TELECOMTEST:SELECT1:MASK:ELEMENT1:WCOUNT 1,2
:GONOGO:TELECOMTEST:SELECT1:MASK:ELEMENT1:WCOUNT? ->
:GONOGO:TELECOMTEST:SELECT1:MASK:ELEMENT1:WCOUNT 2.000E+00,1.000E+00
```

:GONogo:ZPARameter?

Function Queries all settings related to zone/parameter determination.

Syntax
```
:GONogo:ZPARameter?
```

Example
```
:GONOGO:ZPARAMETER? ->
:GONOGO:ZPARAMETER:SELECT1:MODE PARAMETER;PARAMETER:CATEGORY FFT;FFT1:CALCULATION1 0.000E+00,1.000E+00;:GONOGO:ZPARAMETER:SELECT1:RECTANGLE:HORIZONTAL 0.000E+00,1.000E+00;VERTICAL 0.000E+00,1.000E+00;:GONOGO:ZPARAMETER:SELECT1:TRACE 1;WAVE:TRANGE 1.000E+00,2.000E+00;:GONOGO:ZPARAMETER:SELECT1:WINDOW MAIN;:GONOGO:ZPARAMETER:SELECT2:MODE PARAMETER;PARAMETER:CATEGORY FFT;;FFT1:CALCULATION1 0.000E+00,1.000E+00;:GONOGO:ZPARAMETER:SELECT2:RECTANGLE:HORIZONTAL 0.000E+00,1.000E+00;VERTICAL 0.000E+00,1.000E+00;:GONOGO:ZPARAMETER:SELECT2:TRACE 1;WAVE:TRANGE 1.000E+00,2.000E+00;:GONOGO:ZPARAMETER:SELECT2:WINDOW MAIN;:GONOGO:ZPARAMETER:SELECT3:MODE PARAMETER;PARAMETER:CATEGORY FFT;FFT1:CALCULATION1 0.000E+00,1.000E+00;:.....
```

:GONogo:ZPARameter:SELect<x>?

Function Queries all settings related to the condition of the zone/parameter determination.

Syntax
```
:GONogo:ZPARameter:SELect<x>?
```
<x> = 1 to 4

Example
```
:GONOGO:ZPARAMETER:SELECT1? ->
:GONOGO:ZPARAMETER:SELECT1:MODE PARAMETER;PARAMETER:CATEGORY FFT;FFT1:CALCULATION1 0.000E+00,1.000E+00;:GONOGO:ZPARAMETER:SELECT1:RECTANGLE:HORIZONTAL 0.000E+00,1.000E+00;VERTICAL 0.000E+00,1.000E+00;:GONOGO:ZPARAMETER:SELECT1:TRACE 1;WAVE:TRANGE 1.000E+00,2.000E+00;:GONOGO:ZPARAMETER:SELECT1:WINDOW MAIN
```

:GONogo:ZPARameter:SELect<x>:MODE

Function Sets the mode of the condition or queries the current setting.

Syntax
```
:GONogo:ZPARameter:SELect<x>:MODE {PARameter|POLYgon|RECTangle|WAVE}
:GONogo:ZPARameter:SELect<x>:MODE?
```
<x> = 1 to 4

Example
```
:GONOGO:ZPARAMETER:SELECT1:MODE WAVE
:GONOGO:ZPARAMETER:SELECT1:MODE? ->
:GONOGO:ZPARAMETER:SELECT1:MODE WAVE
```

:GONogo:ZPARameter:SELect<x>:PARameter?

Function Queries all settings related to the condition parameter.

Syntax `:GONogo:ZPARameter:SELect<x>:PARameter?`
`<x>` = 1 to 4

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER?
-> :GONOGO:ZPARAMETER:SELECT1:
PARAMETER:CATEGORY FFT;PARAMETER:
CATEGORY FFT;FFT1:
CALCULATION1 1.000E+00,0.000E+00
```

:GONogo:ZPARameter:SELect<x>:PARameter:CATegory

Function Sets the parameter category or queries the current setting.

Syntax `:GONogo:ZPARameter:SELect<x>:PARameter:CATegory {FFT|MEASure|XY}`
`:GONogo:ZPARameter:SELect<x>:PARameter:CATegory?`
`<x>` = 1 to 4

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
CATEGORY FFT
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
CATEGORY? -> :GONOGO:ZPARAMETER:
SELECT1:PARAMETER:CATEGORY FFT
```

Description This command is valid when `:MEASURE:MODE CYCLE`.

:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>?

Function Queries all settings related to the FFT determination.

Syntax `:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>?`
`<x>` of SELect<x> = 1 to 4
`<x>` of FFT<x> = 1 or 2

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
FFT1? -> :GONOGO:ZPARAMETER:SELECT1:
PARAMETER:FFT1:PEAK:
FREQUENCY1 0.000E+00,0.000E+00
```

:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:CALCulation<x>

Function Sets the upper and lower limits of the calculation item of the FFT determination or queries the current setting.

Syntax `:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:CALCulation<x> {<NRf>,<NRf>}`
`:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:CALCulation<x>?`
`<x>` of SELect<x> = 1 to 4
`<x>` of FFT<x> = 1 or 2
`<x>` of CALCulation<x> = 1 to 4
<NRf> = –4 to 4 (div)

Example
```
GONOGO:ZPARAMETER:SELECT1:PARAMETER:
FFT1:CALCULATION1 0,1
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
FFT1:CALCULATION1? ->
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
FFT1:CALCULATION1 1.000E+00,0.000E+00
```

:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:PEAK?

Function Queries all settings related to the peak value of the FFT determination.

Syntax `:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:PEAK?`
`<x>` of SELect<x> = 1 to 4
`<x>` of FFT<x> = 1 or 2

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
FFT1:PEAK? -> :GONOGO:ZPARAMETER:
SELECT1:PARAMETER:FFT1:PEAK:
FREQUENCY1 1.000E+00,0.000E+00
```

:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:PEAK:DFREquency

Function Sets the upper and lower limits between the peak frequencies of the FFT determination or queries the current setting.

Syntax `:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:PEAK:DFREquency {<Frequency>,<Frequency>}`
`:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:PEAK:DFREquency?`
`<x>` of SELect<x> = 1 to 4
`<x>` of FFT<x> = 1 or 2
<Frequency> = See the DL9000 User's Manual

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
FFT1:PEAK:DFREQUENCY 0,1
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
FFT1:PEAK:DFREQUENCY? -> :GONOGO:
ZPARAMETER:SELECT1:PARAMETER:FFT1:PEAK:
DFREQUENCY 1.000E+00,0.000E+00
```

:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:PEAK:DV

Function Sets the upper and lower limits between the peak voltages of the FFT determination or queries the current setting.

Syntax
```
:GONogo:ZPARameter:SELect<x>:PARameter:
FFT<x>:PEAK:DV {<NRf>,<NRf>}
:GONogo:ZPARameter:SELect<x>:PARameter:
FFT<x>:PEAK:DV?
```
<x> of SELect<x> = 1 to 4
<x> of FFT<x> = 1 or 2
<NRf> = –4 to 4 (div)

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
FFT1:PEAK:DV 0,1
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
FFT1:PEAK:DV? -> :GONOGO:ZPARAMETER:
SELECT1:PARAMETER:FFT1:PEAK:
DV 1.000E+00,0.000E+00
```

:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:PEAK:FREQuency<x>

Function Sets the upper and lower limits of the peak frequency of the FFT determination or queries the current setting.

Syntax
```
:GONogo:ZPARameter:SELect<x>:PARameter:
FFT<x>:PEAK:FREQuency<x>
{<Frequency>,<Frequency>}
:GONogo:ZPARameter:SELect<x>:PARameter:
FFT<x>:PEAK:FREQuency<x>?
```
<x> of SELect<x> = 1 to 4
<x> of FFT<x> = 1 or 2
<x> of FREQuency<x> = 1 or 2
<Frequency> = See the DL9000 User's Manual

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
FFT1:PEAK:FREQUENCY1 0,1
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
FFT1:PEAK:FREQUENCY1? -> :GONOGO:
ZPARAMETER:SELECT1:PARAMETER:FFT1:
PEAK:FREQUENCY1 1.000E+00,0.000E+00
```

:GONogo:ZPARameter:SELect<x>:PARameter:FFT<x>:PEAK:V<x>

Function Sets the upper and lower limits of the peak voltage of the FFT determination or queries the current setting.

Syntax
```
:GONogo:ZPARameter:SELect<x>:PARameter:
FFT<x>:PEAK:V<x> {<NRf>,<NRf>}
:GONogo:ZPARameter:SELect<x>:PARameter:
FFT<x>:PEAK:V<x>?
```
<x> of SELect<x> = 1 to 4
<x> of FFT<x> = 1 or 2
<x> of V<x> = 1 or 2
<NRf> = –4 to 4 (div)

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
FFT1:PEAK:V1 1,2
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
FFT1:PEAK:V1? -> :GONOGO:ZPARAMETER:
SELECT1:PARAMETER:FFT1:PEAK:
V1 2.000E+00,1.000E+00
```

:GONogo:ZPARameter:SELect<x>:PARameter:MEASure?

Function Queries all settings related to the determination using automated measurement of waveform parameters (measure determination).

Syntax
```
:GONogo:ZPARameter:SELect<x>:PARameter:
MEASure?
```
<x> = 1 to 4

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
MEASURE? -> :GONOGO:ZPARAMETER:
SELECT1:PARAMETER:MEASURE:
CALCULATION1 2.000E+00,1.000E+00
```

:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:CALCulation<x>

Function Sets the upper and lower limits of the calculation item of the measure determination or queries the current setting.

Syntax
```
:GONogo:ZPARameter:SELect<x>:PARameter:
MEASure:CALCulation<x> {<NRf>,<NRf>}
:GONogo:ZPARameter:SELect<x>:PARameter:
MEASure:CALCulation<x>?
```
<x> of SELect<x> = 1 to 4
<x> of CALCulation<x> = 1 to 4
<NRf> = –4 to 4 (div)

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
MEASURE:CALCULATION1 1,2
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:
MEASURE:CALCULATION1? -> :GONOGO:
ZPARAMETER:SELECT1:PARAMETER:MEASURE:
CALCULATION1 2.000E+00,1.000E+00
```

:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:STATistics

Function Sets the statistical value of the measure determination or queries the current setting.

Syntax
```
:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:STATistics {MAXimum|MEAN|MINimum|SIGMa}
:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:STATistics?
```
<x> = 1 to 4

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:MEASURE:STATISTICS MAXIMUM
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:MEASURE:STATISTICS? ->
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:MEASURE:STATISTICS MAXIMUM
```

Description This command is valid when `:MEASURE:MODE CYCLE`.

:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:TRACe<x>?

Function Queries all settings related to the trace of the measure determination.

Syntax
```
:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:TRACe<x>?
```
<x> of SELect<x> = 1 to 4
<x> of TRACe<x> = 1 to 8

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:MEASURE:TRACE1? ->
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:MEASURE:TRACE1:AREA1:TYPE:MAXIMUM 2.000E+00,1.000E+00
```

:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>?

Function Queries all settings related to the area of the measure determination.

Syntax
```
:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>?
```
<x> of SELect<x> = 1 to 4
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:MEASURE:TRACE1:AREA1? ->
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:MEASURE:TRACE1:AREA1:TYPE:MAXIMUM 2.000E+00,1.000E+00
```

:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>:TYPE?

Function Queries all settings related to the waveform parameters of the measure determination.

Syntax
```
:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>:TYPE?
```
<x> of SELect<x> = 1 to 4
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:MEASURE:TRACE1:AREA1:TYPE? ->
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:MEASURE:TRACE1:AREA1:TYPE:MAXIMUM 2.000E+00,1.000E+00
```

:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>:TYPE:<Parameter>

Function Sets the upper and lower limits of the waveform parameter of the measure determination or queries the current setting.

Syntax
```
:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>:TYPE:<Parameter>
{<NRf>,<NRf>|<Voltage>,<Voltage>|<Current>,<Current>|<Time>,<Time>|<Frequency>,<Frequency>}
:GONogo:ZPARameter:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>:TYPE:<Parameter>?
```
<x> of SELect<x> = 1 to 4
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2
<Parameter> = {BURSt|CMEan|COUNt|CRMS|CSDeviation|DELay|DUTYcycle|FALL|FREQuency|HIGH|HILow|LOW|MAXimum|MEAN|MINimum|NOVershoot|NWIDth|PERFrequency|PERiod|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEViation|TYCInteg|TYINteg}
<NRf>, <Voltage>, <Current>, <Time>, and <Frequency> = See the DL9000 User's Manual.

Example (The following is an example for the maximum value of trace 1 and area 1.)
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:MEASURE:TRACE1:AREA1:TYPE:MAXIMUM 1,2
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:MEASURE:TRACE1:AREA1:TYPE:MAXIMUM? ->
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:MEASURE:TRACE1:AREA1:TYPE:MAXIMUM 2.000E+00,1.000E+00
```

Description This command applies to cycle statistical processing when `:MEASURE:MODE CYCLE`.

:GONogo:ZPARameter:SELect<x>:PARameter:XY<x>?

Function Queries all settings related to the XY determination.

Syntax
```
:GONogo:ZPARameter:SELect<x>:PARameter:XY<x>?
```
<x> of SELect<x> = 1 to 4
<x> of XY<x> = 1 or 2

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:XY1? -> :GONOGO:ZPARAMETER:SELECT1:PARAMETER:XY1:
XYINTEG 2.000E+00,1.000E+00
```

:GONogo:ZPARameter:SELect<x>:PARameter:XY<x>:XYINteg

Function Sets the upper and lower limits integral value of the XY determination or queries the current setting.

Syntax
```
:GONogo:ZPARameter:SELect<x>:PARameter:XY<x>:XYINteg {<NRf>,<NRf>}
:GONogo:ZPARameter:SELect<x>:PARameter:XY<x>:XYINteg?
```
<x> of SELect<x> = 1 to 4
<x> of XY<x> = 1 or 2
<NRf> = –4 to 4 (div)

Example
```
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:XY1:XYINTEG 1,2
:GONOGO:ZPARAMETER:SELECT1:PARAMETER:XY1:XYINTEG? -> :GONOGO:ZPARAMETER:SELECT1:PARAMETER:XY1:
XYINTEG 2.000E+00,1.000E+00
```

:GONogo:ZPARameter:SELect<x>:RECTangle?

Function Queries all settings related to the rectangle determination.

Syntax
```
:GONogo:ZPARameter:SELect<x>:RECTangle?
```
<x> = 1 to 4

Example
```
:GONOGO:ZPARAMETER:SELECT1:RECTANGLE? -> :GONOGO:ZPARAMETER:SELECT1:RECTANGLE:HORIZONTAL 1.000E+00,
0.000E+00;VERTICAL 1.000E+00,0.000E+00
```

:GONogo:ZPARameter:SELect<x>:RECTangle:HORizontal

Function Sets the horizontal position of the rectangle used in the rectangle determination or queries the current setting.

Syntax
```
:GONogo:ZPARameter:SELect<x>:RECTangle:HORizontal {<NRf>,<NRf>}
:GONogo:ZPARameter:SELect<x>:RECTangle:HORizontal?
```
<x> = 1 to 4
<NRf> = –5 to 5 div

Example
```
:GONOGO:ZPARAMETER:SELECT1:RECTANGLE:HORIZONTAL 0,1
:GONOGO:ZPARAMETER:SELECT1:RECTANGLE:HORIZONTAL? -> :GONOGO:ZPARAMETER:SELECT1:RECTANGLE:
HORIZONTAL 1.000E+00,0.000E+00
```

:GONogo:ZPARameter:SELect<x>:RECTangle:VERTical

Function Sets the vertical position of the rectangle used in the rectangle determination or queries the current setting.

Syntax
```
:GONogo:ZPARameter:SELect<x>:RECTangle:VERTical {<NRf>,<NRf>}
:GONogo:ZPARameter:SELect<x>:RECTangle:VERTical?
```
<x> = 1 to 4
<NRf> = –4 to 4 (div)

Example
```
:GONOGO:ZPARAMETER:SELECT1:RECTANGLE:VERTICAL 0,1
:GONOGO:ZPARAMETER:SELECT1:RECTANGLE:VERTICAL? -> :GONOGO:ZPARAMETER:SELECT1:RECTANGLE:
VERTICAL 1.000E+00,0.000E+00
```

:GONogo:ZPARameter:SELect<x>:TRACe

Function Sets the source trace of the zone/parameter determination or queries the current setting.

Syntax
```
:GONogo:ZPARameter:SELect<x>:TRACe {<NRf>}
:GONogo:ZPARameter:SELect<x>:TRACe?
```
<x> = 1 to 4
<NRf> = 1 to 8

Example
```
:GONOGO:ZPARAMETER:SELECT1:TRACE 1
:GONOGO:ZPARAMETER:SELECT1:TRACE? ->
:GONOGO:ZPARAMETER:SELECT1:TRACE 1
```

:GONogo:ZPARameter:SELect<x>:WAVE?

Function Queries all settings related to the wave determination.

Syntax
```
:GONogo:ZPARameter:SELect<x>:WAVE?
```
<x> = 1 to 4

Example
```
:GONOGO:ZPARAMETER:SELECT1:WAVE? ->
:GONOGO:ZPARAMETER:SELECT1:WAVE:
TRANGE 2.000E+00,1.000E+00
```

:GONogo:ZPARameter:SELect<x>:WAVE:EDIT<x>:EXIT

Function Exits the edit menu of the wave determination zone.

Syntax `:GONogo:ZPARameter:SELect<x>:WAVE:EDIT<x>:EXIT {<NRf>|QUIT}`
<x> of SELect<x> = 1 to 4
<x> of EDIT<x> = 1 to13 (1 to 8 are traces and 9 to 13 are internal memories)
<NRf> = 1 to 4 (internal memories)

Example `:GONOGO:ZPARAMETER:SELECT1:WAVE:EDIT1:EXIT 1`

Description
- An error occurs if this command is issued when the zone is not being edited.
- Saves the zone waveform to the internal memory specified by <NRf>.

:GONogo:ZPARameter:SELect<x>:WAVE:EDIT<x>:PART

Function Editing the portion of the zone of the wave determination.

Syntax `:GONogo:ZPARameter:SELect<x>:WAVE:EDIT<x>:PART {<NRf>,<NRf>,<NRf>,<NRf>}`
<x> of SELect<x> = 1 to 4
<x> of EDIT<x> = 1 to13 (1 to 8 are traces. 9 to 12 are internal memories. 13 is the zone waveform that is currently displayed)
<NRf> = –5 to 5 (div: partial editing time axis cursor 1 and 2)
–8 to 8 (div: up and down) (in the order partial editing time axis cursor 1, 2, ↑, and ↓)

Example `:GONOGO:ZPARAMETER:SELECT1:WAVE:EDIT1:PART 1,2,3,4`

Description The amount of movement upward or downward is a relative value with respect to the zone waveform that is currently displayed.

:GONogo:ZPARameter:SELect<x>:WAVE:EDIT<x>:WHOLe

Function Sets the editing of the entire zone of the waveform zone.

Syntax `:GONogo:ZPARameter:SELect<x>:WAVE:EDIT<x>:WHOLe {<NRf>,<NRf>,<NRf>,<NRf>}`
<x> of SELect<x> = 1 to 4
<x> of EDIT<x> = 1 to13 (1 to 8 are traces. 9 to 12 are internal memories. 13 is the zone waveform that is currently displayed)
<NRf> = 0 to 5 (div: left and right)
0 to 8 (div: up and down)
(In the order ←, →, ↑, and ↓)

Example `:GONOGO:ZPARAMETER:SELECT1:WAVE:EDIT1:WHOLE 1,2,3,4`

Description The amount of movement is a relative value with respect to the reference waveform.

:GONogo:ZPARameter:SELect<x>:WAVE:TRANge

Function Sets the determination range of the zone determination or queries the current setting.

Syntax `:GONogo:ZPARameter:SELect<x>:WAVE:TRANge {<NRf>,<NRf>}`
`:GONogo:ZPARameter:SELect<x>:WAVE:TRANge?`
<x> = 1 to 4
<NRf> = –5 to 5 (div)

Example
```
:GONOGO:ZPARAMETER:SELECT1:WAVE:TRANGE 1,2
:GONOGO:ZPARAMETER:SELECT1:WAVE:TRANGE?
-> :GONOGO:ZPARAMETER:SELECT1:WAVE:TRANGE 2.000E+00,1.000E+00
```

:GONogo:ZPARameter:SELect<x>:WINDow

Function Sets the source window of the zone determination or queries the current setting.

Syntax `:GONogo:ZPARameter:SELect<x>:WINDow {MAIN|XY1|XY2|Z1|Z2}`
`:GONogo:ZPARameter:SELect<x>:WINDow?`
<x> = 1 to 4

Example
```
:GONOGO:ZPARAMETER:SELECT1:WINDOW MAIN
:GONOGO:ZPARAMETER:SELECT1:WINDOW? ->
:GONOGO:ZPARAMETER:SELECT1:WINDOW MAIN
```

5.13 HCOPy Group

:HCOPy? (Hard COPY)

Function Queries all settings related to the output of screen data.

Syntax :HCOPy?

Example
```
:HCOPY? -> :HCOPY:DIRECTION EXTPRINTER;
EXTPRINTER:TONE 1;TYPE EINKJET;:HCOPY:
FILE:FORMAT BMP;SAVE:ANAMING DATE;
CDIRECTORY "\Flash Mem\DIR1\DIR2";
NAME "SAMPLE";:HCOPY:FILE:TONE COLOR;:
HCOPY:PRINTER:HRMODE 1
```

:HCOPy:ABORt

Function Aborts data output and paper feeding.

Syntax `:HCOPy:ABORt`

Example `:HCOPY:ABORT`

Description This command is valid for :HCOPy:DIRection EXTPrinter|PRINter.

:HCOPy:DIRection

Function Sets the data output destination or queries the current setting.

Syntax
```
:HCOPy:DIRection {EXTPrinter|FILE|
PRINter}
:HCOPy:DIRection?
```

Example
```
:HCOPY:DIRECTION EXTPRINTER
:HCOPY:DIRECTION? -> :HCOPY:DIRECTION
EXTPRINTER
```

:HCOPy:EXECute

Function Executes the data output. This is an overlap command.

Syntax
```
:HCOPy:EXECute {<String>}
```
<String> = Up to 259 characters

Example
```
:HCOPY:EXECUTE
:HCOPY:EXECUTE "\Flash Mem\DIR\DATA"
```
(Absolute path designation)
```
:HCOPY:EXECUTE "DATA"
```
(Relative path designation)

Description
- If the path is not specified, the data is saved to the file name specified by `:HCOPy:FILE:SAVE:NAME`.
- If the path is specified, the auto naming function is disabled.
- For details on the <String> parameter, see the description in `:HCOPy:FILE:SAVE:CDIRectory`.

:HCOPy:EXTPrinter?

Function Queries all settings related to the external printer output.

Syntax `:HCOPy:EXTPrinter?`

Example
```
:HCOPY:EXTPRINTER? ->
:HCOPY:EXTPRINTER:TONE 1;TYPE EINKJET
```

: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 {EINKjet|
HINKjet}
:HCOPy:EXTPrinter:TYPE?
```

Example
```
:HCOPY:EXTPRINTER:TYPE EINKJET
:HCOPY:EXTPRINTER:TYPE? ->
:HCOPY:EXTPRINTER:TYPE EINKJET
```

:HCOPy:FILE?

Function Queries all settings related to file output.

Syntax `:HCOPy:FILE?`

Example
```
:HCOPY:FILE? -> :HCOPY:FILE:FORMAT BMP;
SAVE:ANAMING DATE;
CDIRECTORY "\Flash Mem\DIR1\DIR2";
NAME "SAMPLE";:HCOPY:FILE:TONE COLOR
```

:HCOPy:FILE:FORMat

Function Sets the file output image format or queries the current setting.

Syntax
```
:HCOPy:FILE:FORMat {BMP|JPEG|PNG}
:HCOPy:FILE:FORMat?
```

Example
```
:HCOPY:FILE:FORMAT BMP
:HCOPY:FILE:FORMAT? ->
:HCOPY:FILE:FORMAT BMP
```

:HCOPy:FILE:SAVE?

Function Queries all settings related to the saving of file output.

Syntax `:HCOPy:FILE:SAVE?`

Example
```
:HCOPY:FILE:SAVE? ->
:HCOPY:FILE:SAVE:ANAMING DATE;
CDIRECTORY "\Flash Mem\DIR1\DIR2";
NAME "SAMPLE"
```

:HCOPy:FILE:SAVE:ANAMing

Function Sets the type of auto naming of save destination file names or queries the current setting.

Syntax
```
:HCOPy:FILE:SAVE:ANAMing {DATE|NUMBering|OFF}
:HCOPy:FILE:SAVE:ANAMing?
```

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

:HCOPy:FILE:SAVE:CDIRectory (Change Directory)

Function Sets the save destination directory name or queries the current setting.

Syntax
```
:HCOPy:FILE:SAVE:CDIRectory {<String>}
:HCOPy:FILE:SAVE:CDIRectory?
```
<String> = Up to 259 characters

Example
```
:HCOPY:FILE:SAVE:CDIRECTORY "\Flash Mem\DIR1\DIR2"
```
(Absolute path designation)
```
:HCOPY:FILE:SAVE:CDIRECTORY "DIR2"
```
(Relative path designation)
```
:HCOPY:FILE:SAVE:CDIRECTORY "\"
```
(Root directory designation)
```
:HCOPY:FILE:SAVE:CDIRECTORY? ->
:HCOPY:FILE:SAVE:
CDIRECTORY "\Flash Mem\DIR1\DIR2"
```

Description
- Data files cannot be saved to the root directory. Specify a save destination drive for the current directory.
  The following five drives are selectable.
  - Internal hard disk: “HD”
  - Internal memory: “Flash Mem”
  - Network drive: “Network”
  - PC card: “Storage Card<x>”
  - USB storage device: “USB Storage<x>”

  <x> = 1 to 4 (however, only a total of four “Storage Card” and “USB Storage” designations are supported (including partitions))
- Sets the specified directory the current directory for saving and loading.
- Absolute and relative path designations are possible.
- To specify an absolute path, enter a backslash at the front of the path.
- Relative path to higher level directories is not allowed.

:HCOPy:FILE:SAVE:NAME

Function Sets the save destination file name or queries the current setting.

Syntax
```
:HCOPy:FILE:SAVE:NAME <Filename>
:HCOPy:FILE:SAVE:NAME?
```

Example
```
:HCOPY:FILE:SAVE:NAME "SAMPLE"
:HCOPY:FILE:SAVE:NAME? ->
:HCOPY:FILE:SAVE:NAME "SAMPLE"
```

:HCOPy:PRINter?

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

Syntax
```
:HCOPy:PRINter?
```

Example
```
:HCOPY:PRINTER? ->
:HCOPY:PRINTER:HRMODE 1
```

:HCOPy:PRINter:HRMode

Function Turns ON/OFF the high resolution mode of the built-in printer output or queries the current setting.

Syntax
```
:HCOPy:PRINter:HRMode {<Boolean>}
:HCOPy:PRINter:HRMode?
```

Example
```
:HCOPY:PRINTER:HRMODE ON
:HCOPY:PRINTER:HRMODE? ->
:HCOPY:PRINTER:HRMODE 1
```

5.14 HISTory Group

:HISTory?

Function Queries all settings related to the history function.

Syntax :HISTory?

Example
```
:HISTORY? -> :HISTORY:CURRENT:
DISPLAY 0,-10;DMODE ONE;MODE RECORD;
RECORD 0;REPLAY:SPEED 1;:HISTORY:
CURRENT:SEARCH:LOGIC AND;SELECT1:
CONDITION IN;MODE PARAMETER;PARAMETER:
CATEGORY MEASURE;MEASURE:
CALCULATION1 1.000E+00,2.000E+00;:
HISTORY:CURRENT:SEARCH:SELECT1:
RECTANGLE:HORIZONTAL 1.000E+00,
2.000E+00;VERTICAL 1.000E+00,
2.000E+00;:HISTORY:CURRENT:SEARCH:
SELECT1:TRACE 1;:HISTORY:CURRENT:
SEARCH:SELECT1:WAVE:TRANGE 1.000E+00,
2.000E+00;:HISTORY:CURRENT:SEARCH:
SELECT1:WINDOW MAIN;:HISTORY:CURRENT:
SEARCH:SELECT2:CONDITION IN;
MODE PARAMETER;PARAMETER:
CATEGORY MEASURE;MEASURE:
CALCULATION1 1.000E+00,2.000E+00;:
HISTORY:CURRENT:SEARCH:SELECT2:
RECTANGLE:HORIZONTAL 1.000E+00,
2.000E+00;VERTICAL 1.000E+00,
2.000E+00;:HISTORY:CURRENT:SEARCH:
SELECT2:TRACE 1;:HISTORY:CURRENT:
SEARCH:SELECT2:WAVE:TRANGE 1.000E+00,
2.000E+00;:HISTORY:CURRENT:SEARCH:
SELECT2:WINDOW MAIN;:HISTORY:CURRENT:
SEARCH;SELECT3:CONDITION IN;
MODE PARAMETER;PARAMETER:
CATEGORY MEASURE;MEASURE:
CALCULATION1 1.000E+00,2.000E+00......
```

:HISTory[:CURRent]?

Function Queries all settings related to the history function of the current waveform (CH1 to 4, M1 to 8).

Syntax :HISTory[:CURRent]?

Example
```
:HISTORY:CURRENT? -> :HISTORY:CURRENT:
DISPLAY 0,-10;DMODE ONE;MODE RECORD;
RECORD 0;REPLAY:SPEED 1;:HISTORY:
CURRENT:SEARCH:LOGIC AND;SELECT1:
CONDITION IN;MODE PARAMETER;PARAMETER:
CATEGORY MEASURE;MEASURE:
CALCULATION1 1.000E+00,2.000E+00;:
HISTORY:CURRENT:SEARCH:SELECT1:
RECTANGLE:HORIZONTAL 1.000E+00,
2.000E+00;VERTICAL 1.000E+00,
2.000E+00;:HISTORY:CURRENT:SEARCH:
SELECT1:TRACE 1;:HISTORY:CURRENT:
SEARCH:SELECT1:WAVE:TRANGE 1.000E+00,
2.000E+00;:HISTORY:CURRENT:SEARCH:
SELECT1:WINDOW MAIN;:HISTORY:CURRENT:
SEARCH:SELECT2:CONDITION IN;
MODE PARAMETER;PARAMETER:
CATEGORY MEASURE;MEASURE:
CALCULATION1 1.000E+00,2.000E+00;:
HISTORY:CURRENT:SEARCH:SELECT2:
RECTANGLE:HORIZONTAL 1.000E+00,
2.000E+00;VERTICAL 1.000E+00,
2.000E+00;:HISTORY:CURRENT:SEARCH:
SELECT2:TRACE 1;:HISTORY:CURRENT:
SEARCH:SELECT2:WAVE:TRANGE 1.000E+00,
2.000E+00;:HISTORY:CURRENT:SEARCH:
SELECT2:WINDOW MAIN;:HISTORY:CURRENT:
SEARCH;SELECT3:CONDITION IN;
MODE PARAMETER;PARAMETER:
CATEGORY MEASURE;MEASURE:
CALCULATION1 1.000E+00,2.000E+00......
```

:HISTory[:CURRent]:DISPlay

Function Sets the start number and end number of the display record of the history waveform or queries the current setting.

Syntax
```
:HISTory[:CURRent]:DISPlay
{<NRf>,<NRf>}
:HISTory[:CURRent]:DISPlay?
```
<NRf> = See the DL9000 User's Manual.

Example
```
:HISTORY:CURRENT:DISPLAY 0,-10
:HISTORY:CURRENT:DISPLAY? ->
:HISTORY:CURRENT:DISPLAY 0,-10
```

:HISTory[:CURRent]:DMODe (Display Mode)

Function Sets the display mode of the history waveform or queries the current setting.

Syntax
```
:HISTory[:CURRent]:DMODe {ACOLor|AHTone|AINTensity|ONE}
:HISTory[:CURRent]:DMODe?
```

Example
```
:HISTORY:CURRENT:DMODE ONE
:HISTORY:CURRENT:DMODE? ->
:HISTORY:CURRENT:DMODE ONE
```

:HISTory[:CURRent]:MODE

Function Sets the highlight display mode of the history waveform or queries the current setting.

Syntax
```
:HISTory[:CURRent]:MODE {AVERage|RECord}
:HISTory[:CURRent]:MODE?
```

Example
```
:HISTORY:CURRENT:MODE RECORD
:HISTORY:CURRENT:MODE? ->
:HISTORY:CURRENT:MODE RECORD
```

:HISTory[:CURRent]:RECord

Function Sets the target record of the history waveform or queries the current setting.

Syntax
```
:HISTory[:CURRent]:RECord {<NRf>|MINimum}
:HISTory[:CURRent]:RECord?
```
<NRf> = See the DL9000 User's Manual.

Example
```
:HISTORY:CURRENT:RECORD 0
:HISTORY:CURRENT:RECORD? ->
:HISTORY:CURRENT:RECORD 0
```

Description Specifying `MINimum` sets the record to the minimum record number.

:HISTory[:CURRent]:RECord? MINimum

Function Queries the minimum record number of the history waveform.

Syntax
```
:HISTory[:CURRent]:RECord? MINimum
```

Example
```
:HISTORY:CURRENT:RECORD? MINIMUM ->
:HISTORY:CURRENT:RECORD -1
```

Description Specifying `MINimum` sets the record to the minimum record number.

:HISTory[:CURRent]:REPLay?

Function Queries all settings related to the replay function of the history function.

Syntax
```
:HISTory[:CURRent]:REPLay?
```

Example
```
:HISTORY:CURRENT:REPLAY? ->
:HISTORY:CURRENT:REPLAY:SPEED 1
```

:HISTory[:CURRent]:REPLay:JUMP

Function Jumps the history waveform to the specified record number.

Syntax
```
:HISTory[:CURRent]:REPLay:JUMP {MAXimum|MINimum}
```

Example
```
:HISTORY:CURRENT:REPLAY:JUMP MAXIMUM
```

:HISTory[:CURRent]:REPLay:SPEed

Function Sets the replay speed of the history waveform or queries the current setting.

Syntax
```
:HISTory[:CURRent]:REPLay:SPEed {<NRf>|PER3|PER10|PER30|PER60}
:HISTory[:CURRent]:REPLay:SPEed?
```
<NRf> = 1, 3, 10

Example
```
:HISTORY:CURRENT:REPLAY:SPEED 1
:HISTORY:CURRENT:REPLAY:SPEED? ->
:HISTORY:CURRENT:REPLAY:SPEED 1
```

:HISTory[:CURRent]:REPLay:STARt

Function Starts the replay of the history waveform in the specified direction.

Syntax
```
:HISTory[:CURRent]:REPLay:STARt {MAXimum|MINimum}
```

Example
```
:HISTORY:CURRENT:REPLAY:START MAXIMUM
```

:HISTory[:CURRent]:REPLay:STOP

Function Stops the replay of the history waveform.

Syntax
```
:HISTory[:CURRent]:REPLay:STOP
```

Example
```
:HISTORY:CURRENT:REPLAY:STOP
```

:HISTory[:CURRent][:SEARch]?

Function Queries all settings related to the history search function.

Syntax :HISTory[:CURRent][:SEARch]?

Example :HISTORY:CURRENT:SEARCH? ->
:HISTORY:CURRENT:SEARCH:LOGIC AND;
SELECT1:CONDITION IN;MODE PARAMETER;
PARAMETER:CATEGORY MEASURE;MEASURE:
CALCULATION1 1.000E+00,2.000E+00;:
HISTORY:CURRENT:SEARCH:SELECT1:
RECTANGLE:HORIZONTAL 1.000E+00,
2.000E+00;VERTICAL 1.000E+00,
2.000E+00;:HISTORY:CURRENT:SEARCH:
SELECT1:TRACE 1;:HISTORY:CURRENT:
SEARCH:SELECT1:WAVE:TRANGE 1.000E+00,
2.000E+00;:HISTORY:CURRENT:SEARCH:
SELECT1:WINDOW MAIN;:HISTORY:CURRENT:
SEARCH:SELECT2:CONDITION IN;
MODE PARAMETER;PARAMETER:
CATEGORY MEASURE;MEASURE:
CALCULATION1 1.000E+00,2.000E+00;:
HISTORY:CURRENT:SEARCH:SELECT2:
RECTANGLE:HORIZONTAL 1.000E+00,
2.000E+00;VERTICAL 1.000E+00,
2.000E+00;:HISTORY:CURRENT:SEARCH:
SELECT2:TRACE 1;:HISTORY:CURRENT:
SEARCH:SELECT2:WAVE:TRANGE 1.000E+00,
2.000E+00;:HISTORY:CURRENT:SEARCH:
SELECT2:WINDOW MAIN;:HISTORY:CURRENT:
SEARCH;SELECT3:CONDITION IN;
MODE PARAMETER;PARAMETER:
CATEGORY MEASURE;MEASURE:
CALCULATION1 1.000E+00,2.000E+00.....

:HISTory[:CURRent][:SEARch]:ABORt

Function Aborts the history search.

Syntax :HISTory[:CURRent][:SEARch]:ABORt

Example :HISTORY:CURRENT:SEARCH:ABORT

:HISTory[:CURRent][:SEARch]:EXECute

Function Executes the history search. This is an overlap command.

Syntax :HISTory[:CURRent][:SEARch]:EXECute

Example :HISTORY:CURRENT:SEARCH:EXECUTE

:HISTory[:CURRent][:SEARch]:LOGic

Function Sets the history search logic or queries the current setting.

Syntax :HISTory[:CURRent][:SEARch]:LOGic {AND|OR}
:HISTory[:CURRent][:SEARch]:LOGic?

Example :HISTORY:CURRENT:SEARCH:LOGIC AND
:HISTORY:CURRENT:SEARCH:LOGIC? ->
:HISTORY:CURRENT:SEARCH:LOGIC AND

:HISTory[:CURRent][:SEARch]:RESet

Function Resets the search conditions of the history search.

Syntax :HISTory[:CURRent][:SEARch]:RESet

Example :HISTORY:CURRENT:SEARCH:RESET

:HISTory[:CURRent][:SEARch]:SELect<x>?

Function Queries all settings related to the history search condition.

Syntax :HISTory[:CURRent][:SEARch]:SELect<x>?
<x> = 1 to 4

Example :HISTORY:CURRENT:SEARCH:SELECT1? ->
:HISTORY:CURRENT:SEARCH:SELECT1:
CONDITION IN;MODE PARAMETER;PARAMETER:
CATEGORY MEASURE;MEASURE:
CALCULATION1 1.000E+00,2.000E+00;:
HISTORY:CURRENT:SEARCH:SELECT1:
RECTANGLE:HORIZONTAL 1.000E+00,
2.000E+00;VERTICAL 1.000E+00,
2.000E+00;:HISTORY:CURRENT:SEARCH:
SELECT1:TRACE 1;:HISTORY:CURRENT:
SEARCH:SELECT1:WAVE:TRANGE 1.000E+00,
2.000E+00;:HISTORY:CURRENT:SEARCH:
SELECT1:WINDOW MAIN

:HISTory[:CURRent][:SEARch]:SELect<x>:CONDition

Function Sets determination criteria of the history search condition or queries the current setting.

Syntax :HISTory[:CURRent][:SEARch]:SELect<x>:
CONDition {DONTcare|IN|OUT}
:HISTory[:CURRent][:SEARch]:SELect<x>:
CONDition?
<x> = 1 to 4

Example :HISTORY:CURRENT:SEARCH:SELECT1:
CONDITION IN
:HISTORY:CURRENT:SEARCH:SELECT1:
CONDITION? -> :HISTORY:CURRENT:SEARCH:
SELECT1:CONDITION IN

:HISTory[:CURRent][:SEARch]:SELect<x>:MODE

Function Sets the mode of the history search condition or queries the current setting.

Syntax :HISTory[:CURRent][:SEARch]:SELect<x>:
MODE {PARameter|POLYgon|RECTangle|WAVE}
:HISTory[:CURRent][:SEARch]:SELect<x>:
MODE?
<x> = 1 to 4

Example :HISTORY:CURRENT:SEARCH:SELECT1:
MODE WAVE
:HISTORY:CURRENT:SEARCH:SELECT1:MODE?
-> :HISTORY:CURRENT:SEARCH:SELECT1:
MODE WAVE

:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter?

Function Queries all settings related to the parameter of the history search condition.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter?
```
<x> = 1 to 4

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:PARAMETER? -> :HISTORY:CURRENT:SEARCH:SELECT1:PARAMETER:CATEGORY MEASURE;MEASURE:CALCULATION1 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:CATegory

Function Sets the parameter category or queries the current setting.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:CATegory {FFT|MEASure|XY}
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:CATegory?
```
<x> = 1 to 4

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:PARAMETER:CATEGORY MEASURE
:HISTORY:CURRENT:SEARCH:SELECT1:PARAMETER:CATEGORY? ->
:HISTORY:CURRENT:SEARCH:SELECT1:PARAMETER:CATEGORY MEASURE
```

:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:FFT<x>?

Function Queries all settings related to the FFT search.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:FFT<x>?
```
<x> of SELect<x> = 1 to 4
<x> of FFT<x> = 1 or 2

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:PARAMETER:FFT1? -> :HISTORY:CURRENT:SEARCH:SELECT1:PARAMETER:FFT1:CALCULATION1 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:FFT<x>:CALCulation<x>

Function Sets the upper and lower limits of the calculation item of the FFT search or queries the current setting.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:FFT<x>:CALCulation<x> {<NRf>,<NRf>}
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:FFT<x>:CALCulation<x>?
```
<x> of SELect<x> = 1 to 4
<x> of FFT<x> = 1 or 2
<x> of CALCulation<x> = 1 to 4
<NRf> = See the DL9000 User's Manual.

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:PARAMETER:FFT1:CALCULATION1 1,2
:HISTORY:CURRENT:SEARCH:SELECT1:PARAMETER:FFT1:CALCULATION1? ->
:HISTORY:CURRENT:SEARCH:SELECT1:PARAMETER:FFT1:CALCULATION1 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:FFT<x>:PEAK?

Function Queries all settings related to the peak value of the FFT search.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:FFT<x>:PEAK?
```
<x> of SELect<x> = 1 to 4
<x> of FFT<x> = 1 or 2

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:PARAMETER:FFT1:PEAK? ->
:HISTORY:CURRENT:SEARCH:SELECT1:PARAMETER:FFT1:PEAK:DV 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]: SELect<x>:PARameter:FFT<x>:PEAK: DFREquency

Function Sets the upper and lower limits between the peak frequencies of the FFT search or queries the current setting.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:FFT<x>:PEAK:DFREquency
{<Frequency>,<Frequency>}
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:FFT<x>:PEAK:DFREquency?
```
<x> of SELect<x> = 1 to 4
<x> of FFT<x> = 1 or 2
<Frequency> = See the DL9000 User's Manual

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:PEAK:DFREQUENCY 1,10
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:PEAK:DFREQUENCY? ->
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:PEAK:
DFREQUENCY 10.00E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]: SELect<x>:PARameter:FFT<x>:PEAK:DV

Function Sets the upper and lower limits between the peak voltages of the FFT search or queries the current setting.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:FFT<x>:PEAK:DV {<NRf>,<NRf>}
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:FFT<x>:PEAK:DV?
```
<x> of SELect<x> = 1 to 4
<x> of FFT<x> = 1 or 2
<NRf> = –4 to 4 (div)

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:PEAK:DV 1,2
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:PEAK:DV? ->
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:PEAK:
DV 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]: SELect<x>:PARameter:FFT<x>:PEAK: FREquency<x>

Function Sets the upper and lower limits of the peak frequency of the FFT search or queries the current setting.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:FFT<x>:PEAK:FREquency<x>
{<Frequency>,<Frequency>}
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:FFT<x>:PEAK:FREquency<x>?
```
<x> of SELect<x> = 1 to 4
<x> of FFT<x> = 1 or 2
<x> of FREQuency<x> = 1 or 2
<Frequency> = See the DL9000 User's Manual

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:PEAK:FREQUENCY1 1,2
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:PEAK:FREQUENCY1? ->
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:PEAK:
FREQUENCY1 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]: SELect<x>:PARameter:FFT<x>:PEAK:V<x>

Function Sets the upper and lower limits of the peak voltage of the FFT search or queries the current setting.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:FFT<x>:PEAK:V<x>
{<NRf>,<NRf>}
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:FFT<x>:PEAK:V<x>?
```
<x> of SELect<x> = 1 to 4
<x> of FFT<x> = 1 or 2
<x> of V<x> = 1 or 2
<NRf> = –4 to 4 (div)

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:PEAK:V1 1,2
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:PEAK:V1? ->
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:FFT1:PEAK:
V1 2.000E+00,1.000E+00
```

SELect<x>:PARameter:MEASure?

Function Queries all settings related to the search using automated measurement of waveform parameters (measure search).

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:MEASure?
```
<x> = 1 to 4

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:MEASURE? ->
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:MEASURE:
CALCULATION1 1.000E+00,2.000E+00
```

:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:MEASure:CALCulation<x>

Function Sets the upper and lower limits of the calculation item of the measure search or queries the current setting.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:MEASure:CALCulation<x>
{<NRf>,<NRf>}
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:MEASure:CALCulation<x>?
```
<x> of SELect<x> = 1 to 4
<x> of CALCulation<x> = 1 to 4
<NRf> = –4 to 4 (div)

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:MEASURE:CALCULATION1 1,2
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:MEASURE:CALCULATION1? ->
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:MEASURE:
CALCULATION1 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:MEASure:TRACe<x>?

Function Queries all settings related to the trace of the measure search.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:MEASure:TRACe<x>?
```
<x> of SELect<x> = 1 to 4
<x> of TRACe<x> = 1 to 8

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:MEASURE:TRACE1? ->
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:MEASURE:TRACE1:AREA1:TYPE:
BURST 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>?

Function Queries all settings related to the area of the measure search.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:MEASure:TRACe<x>:AREA<x>?
```
<x> of SELect<x> = 1 to 4
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:MEASURE:TRACE1:AREA1? ->
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:MEASURE:TRACE1:AREA1:TYPE:
BURST
2.000E+00,1.000E+00SELect<x>:PARameter:
MEASure:
TRACe<x>:AREA<x>:TYPE?
```

:HISTory[:CURRent][:SEARch]:SELect<x>:PARameter:MEASure:TRACe<x>:AREA<x>:TYPE?

Function Queries all settings related to the waveform parameters of the measure search.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:MEASure:TRACe<x>:AREA<x>:
TYPE?
```
<x> of SELect<x> = 1 to 4
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:MEASURE:TRACE1:AREA1:TYPE? ->
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:MEASURE:TRACE1:AREA1:TYPE:
BURST 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]: SELect<x>:PARameter:MEASure: TRACe<x>:AREA<x>:TYPE:<Parameter>

Function Sets the upper and lower limits of the waveform parameter of the measure search or queries the current setting.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:MEASure:TRACe<x>:AREA<x>:
TYPE:<Parameter> {(<NRf>,<NRf>)|
(<Voltage>,<Voltage>)|
(<Current>,<Current>)|(<Time>,<Time>)|
(<Frequency>,<Frequency>)}
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:MEASure:TRACe<x>:AREA<x>:
TYPE:<Parameter>?
```
<x> of SELect<x> = 1 to 4
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2
<Parameter> = {BURSt|CMEan|COUNt|CRMS|CSDeviation|DELay|DUTYcycle|FALL|FREQuency|HIGH|HILow|LOW|MAXimum|MEAN|MINimum|NOVershoot|NWIDth|PERFrequency|PERiod|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEViation|TYCInteg|TYINteg}
<NRf>, <Voltage>, <Current>, <Time>, and <Frequency> = See the DL9000 User's Manual.

Example (The following is an example for the maximum value of trace 1 and area 1.)
```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:MEASURE:TRACE1:AREA1:TYPE:
MAXIMUM 0,1
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:MEASURE:TRACE1:AREA1:TYPE:
MAXIMUM? -> :HISTORY:CURRENT:SEARCH:
SELECT1:PARAMETER:MEASURE:TRACE1:AREA1:
TYPE:MAXIMUM 1.000E+00,0.000E+00
```

:HISTory[:CURRent][:SEARch]: SELect<x>:PARameter:XY<x>?

Function Queries all settings related to the XY search.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:XY<x>?
```
<x> of SELect<x> = 1 to 4
<x> of XY<x> = 1 or 2

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:XY1? ->
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:XY1:
XYINTEG 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]: SELect<x>:PARameter:XY<x>:XYINteg

Function Sets the upper and lower limits integral value of the XY search or queries the current setting.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:XY<x>:XYINteg {<NRf>,<NRf>}
:HISTory[:CURRent][:SEARch]:SELect<x>:
PARameter:XY<x>:XYINteg?
```
<x> of SELect<x> = 1 to 4
<x> of XY<x> = 1 or 2
<NRf> = –4 to 4 (div)

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:XY1:XYINTEG 1,2
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:XY1:XYINTEG? ->
:HISTORY:CURRENT:SEARCH:SELECT1:
PARAMETER:XY1:
XYINTEG 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]: SELect<x>:RECTangle?

Function Queries all settings related to the rectangle search.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:
RECTangle?
```
<x> = 1 to 4

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:
RECTANGLE? -> :HISTORY:CURRENT:SEARCH:
SELECT1:RECTANGLE:HORIZONTAL 2.000E+00,
1.000E+00;VERTICAL 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]: SELect<x>:RECTangle:HORizontal

Function Sets the horizontal position of the rectangle used in the rectangle search or queries the current setting.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:
RECTangle:HORizontal {<NRf>,<NRf>}
:HISTory[:CURRent][:SEARch]:SELect<x>:
RECTangle:HORizontal?
```
<x> = 1 to 4
<NRf> = –5 to 5 (div)

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:
RECTANGLE:HORIZONTAL 1,2
:HISTORY:CURRENT:SEARCH:SELECT1:
RECTANGLE:HORIZONTAL? ->
:HISTORY:CURRENT:SEARCH:SELECT1:
RECTANGLE:HORIZONTAL 2.000E+00,
1.000E+00
```

:HISTory[:CURRent][:SEARch]:SELect<x>:RECTangle:VERTical

Function Sets the vertical position of the rectangle used in the rectangle search or queries the current setting.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:RECTangle:VERTical {<NRf>,<NRf>}
:HISTory[:CURRent][:SEARch]:SELect<x>:RECTangle:VERTical?
```
<x> = 1 to 4
<NRf> = –4 to 4 (div)

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:RECTANGLE:VERTICAL 1,2
:HISTORY:CURRENT:SEARCH:SELECT1:RECTANGLE:VERTICAL? ->
:HISTORY:CURRENT:SEARCH:SELECT1:RECTANGLE:VERTICAL 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]:SELect<x>:TRACe

Function Sets the source trace of the history search or queries the current setting.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:TRACe {<NRf>}
:HISTory[:CURRent][:SEARch]:SELect<x>:TRACe?
```
<x> = 1 to 4
<NRf> = 1 to 8

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:TRACE 1
:HISTORY:CURRENT:SEARCH:SELECT1:TRACE? -> :HISTORY:CURRENT:SEARCH:SELECT1:TRACE 1
```

Description This command is invalid when Window is XY.

:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE?

Function Queries all settings related to the search in the waveform zone (wave search).

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE?
```
<x> = 1 to 4

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:WAVE? -> :HISTORY:CURRENT:SEARCH:SELECT1:WAVE:TRANGE 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE:EDIT<x>:EXIT

Function Exits the zone edit menu of the wave search.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE:EDIT<x>:EXIT {<NRf>|QUIT}
```
<x> of SELect<x> = 1 to 4
<x> of EDIT<x> = 1 to 13
<NRf> = 1 to 4 (internal memories)

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:WAVE:EDIT1:EXIT 1
```

Description
- An error occurs if this command is issued when the zone is not being edited.
- Saves the zone waveform to the internal memory specified by <NRf>.

:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE:EDIT<x>:PART

Function Sets the editing of the portion of the zone of the wave search.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE:EDIT<x>:PART
{<NRf>,<NRf>,<NRf>,<NRf>}
```
<x> of SELect<x> = 1 to 4
<x> of EDIT<x> = 1 to13 (1 to 8 are traces. 9 to 12 are internal memories. 13 is the zone waveform that is currently displayed)
<NRf> = –5 to 5 (div: partial editing time axis cursor 1 and 2)
–8 to 8 (div: up and down) (in the order partial editing time axis cursor 1, 2, ↑, and ↓)

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:WAVE:EDIT1:PART 1,2,3,4
```

Description The amount of movement upward or downward is a relative value with respect to the zone waveform that is currently displayed.

:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE:EDIT<x>:WHOLe

Function Sets the editing of the entire zone of the wave search.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE:EDIT<x>:WHOLe
{<NRf>,<NRf>,<NRf>,<NRf>}
```
<x> of SELect<x> = 1 to 4
<x> of EDIT<x> = 1 to13 (1 to 8 are traces. 9 to 12 are internal memories. 13 is the zone waveform that is currently displayed)
<NRf> = 0 to 5 (div: left and right)
0 to 8 (div: up and down)
(In the order ←, →, ↑, and ↓)

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:WAVE:EDIT1:WHOLE 1,2,3,4
```

Description The amount of movement is a relative value with respect to the reference waveform.

:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE:TRANge

Function Sets the range over which to perform the wave search or queries the current setting.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE:TRANge {<NRf>,<NRf>}
:HISTory[:CURRent][:SEARch]:SELect<x>:WAVE:TRANge?
```
<x> = 1 to 4
<NRf> = –5 to 5 (div)

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:WAVE:TRANGE 1,2
:HISTORY:CURRENT:SEARCH:SELECT1:WAVE:TRANGE? -> :HISTORY:CURRENT:SEARCH:SELECT1:WAVE:TRANGE 2.000E+00,1.000E+00
```

:HISTory[:CURRent][:SEARch]:SELect<x>:WINDow

Function Sets the source window of the history search or queries the current setting.

Syntax
```
:HISTory[:CURRent][:SEARch]:SELect<x>:WINDow {MAIN|XY1|XY2|Z1|Z2}
:HISTory[:CURRent][:SEARch]:SELect<x>:WINDow?
```
<x> = 1 to 4

Example
```
:HISTORY:CURRENT:SEARCH:SELECT1:WINDOW MAIN
:HISTORY:CURRENT:SEARCH:SELECT1:WINDOW? -> :HISTORY:CURRENT:SEARCH:SELECT1:WINDOW MAIN
```

Description XY1 and XY2 are valid only for a rectangular polygon.

:HISTory[:CURRent]:TIME?

Function Queries the time of the source record number of the history waveform.

Syntax
```
:HISTory[:CURRent]:TIME? {<NRf>|MINimum}
```

Example
```
:HISTORY:CURRENT:TIME? -1 ->
:HISTORY:CURRENT:TIME "-1 10:20:30.10"
```

Description Specifying `MINimum` sets the record to the minimum record number.

:HISTory:REFerence<x>?

Function Queries all settings related to the history function of the reference.

Syntax
```
:HISTory:REFerence<x>?
```
<x> = 1 to 4

Example
```
:HISTORY:REFERENCE1? ->
HISTORY:REFERENCE1:DMODE ACOLOR;MODE AVERAGE;RECORD 1;REPLAY:SPEED 1
```

:HISTory:REFerence<x>:DMODe (Display Mode)

Function Sets the display mode of the history waveform of the reference or queries the current setting.

Syntax
```
:HISTory:REFerence<x>:DMODe {ACOLor|AHTone|AINTensity|ONE}
:HISTory:REFerence<x>:DMODe?
```
<x> = 1 to 4

Example
```
:HISTORY:REFERENCE1:DMODE ACOLOR
:HISTORY:REFERENCE1:DMODE? ->
:HISTORY:REFERENCE1:DMODE ACOLOR
```

:HISTory:REFerence<x>:MODE

Function Sets the highlight display mode of the history waveform of the reference or queries the current setting.

Syntax
```
:HISTory:REFerence<x>:MODE {AVERage|RECord}
:HISTory:REFerence<x>:MODE?
```
<x> = 1 to 4

Example
```
:HISTORY:REFERENCE1:MODE AVERAGE
:HISTORY:REFERENCE1:MODE? ->
:HISTORY:REFERENCE1:MODE AVERAGE
```

:HISTory:REFerence<x>:RECord

Function Sets the source record of the history waveform of the reference or queries the current setting.

Syntax
```
:HISTory:REFerence<x>:RECord {<NRf>|MINimum}
:HISTory:REFerence<x>:RECord?
```
<x> = 1 to 4
<NRf> = See the DL9000 User's Manual.

Example
```
:HISTORY:REFERENCE1:RECORD 1
:HISTORY:REFERENCE1:RECORD? ->
:HISTORY:REFERENCE1:RECORD 1
```

:HISTory:REFerence<x>:RECord? MINimum

Function Queries the minimum record number of the history waveform of the reference.

Syntax
```
:HISTory:REFerence<x>:RECord? {MINimum}
```
<x> = 1 to 4

Example
```
:HISTORY:REFERENCE1:RECORD? MINIMUM ->
:HISTORY:REFERENCE1:RECORD -1
```

Description Specifying `MINimum` sets the record to the minimum record number.

:HISTory:REFerence<x>:REPLay?

Function Queries all settings related to the replay function of the history function of the reference.

Syntax
```
:HISTory:REFerence<x>:REPLay?
```
<x> = 1 to 4

Example
```
:HISTORY:REFERENCE1:REPLAY? ->
:HISTORY:REFERENCE1:REPLAY:SPEED 1
```

:HISTory:REFerence<x>:REPLay:JUMP

Function Jumps to the specified record number of the history waveform of the reference.

Syntax `:HISTory:REFerence<x>:REPLay:JUMP {MAXimum|MINimum}`
<x> = 1 to 4

Example `:HISTORY:REFERENCE1:REPLAY:JUMP MAXIMUM`

:HISTory:REFerence<x>:REPLay:SPEed

Function Sets the replay speed of the history waveform of the reference or queries the current setting.

Syntax `:HISTory:REFerence<x>:REPLay:SPEed {<NRf>|PER3|PER10|PER30|PER60}`
`:HISTory:REFerence<x>:REPLay:SPEed?`
<x> = 1 to 4
<NRf> = 1,3,10

Example `:HISTORY:REFERENCE1:REPLAY:SPEED 1`
`:HISTORY:REFERENCE1:REPLAY:SPEED? ->`
`:HISTORY:REFERENCE1:REPLAY:SPEED 1`

:HISTory:REFerence<x>:REPLay:STARt

Function Starts the replay of the history waveform of the reference.

Syntax `:HISTory:REFerence<x>:REPLay:STARt {MAXimum|MINimum}`
<x> = 1 to 4

Example `:HISTORY:REFERENCE1:REPLAY:START MAXIMUM`

:HISTory:REFerence<x>:REPLay:STOP

Function Stops the replay of the history waveform of the reference.

Syntax `:HISTory:REFerence<x>:REPLay:STOP`
<x> = 1 to 4

Example `:HISTORY:REFERENCE1:REPLAY:STOP`

:HISTory:REFerence<x>:TIME?

Function Queries the time of the source record number of the reference waveform.

Syntax `:HISTory:REFerence<x>:TIME? {<NRf>|MINimum}`
<x> = 1 to 4

Example `:HISTORY:REFERENCE1:TIME? -1 ->`
`:HISTORY:REFERENCE1:TIME "-1 10:20:30.10"`

Description Specifying `MINimum` sets the record to the minimum record number.

5.15 IMAGe Group

:IMAGe?

Function Queries all settings related to the output of screen image data.
Syntax :IMAGe?
Example :IMAGE? -> :IMAGE:FORMAT BMP;TONE COLOR

:IMAGe:FORMat

Function Sets the output format of the screen image data or queries the current setting.
Syntax :IMAGe:FORMat {BMP|JPEG|PNG}
:IMAGe:FORMat?
Example :IMAGE:FORMAT BMP
:IMAGE:FORMAT? -> :IMAGE:FORMAT BMP

: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 For details on <Block data>, see page 4-6.

:IMAGe:TONE

Function Sets the tone of the screen image data or queries the current setting.
Syntax :IMAGe:TONE {COLor|GRAY|OFF|REVerse}
:IMAGe:TONE?
Example :IMAGE:TONE COLOR
:IMAGE:TONE? -> :IMAGE:TONE COLOR
Description If ":IMAGe:FORMat JPEG" is specified, OFF cannot be selected.

5.16 INITialize Group

:INITialize:EXECute

Function Execute the initialization.
Syntax :INITialize:EXECute
Example :INITIALIZE:EXECUTE

:INITialize:UNDO

Function Cancels the initialization that was executed.
Syntax :INITialize:UNDO
Example :INITIALIZE:UNDO

5.17 MATH Group

:MATH<x>?

Function Queries all settings related to the computation.

Syntax
```
:MATH<x>?
```
<x> = 1 to 8

Example
```
:MATH1? -> :MATH1:SELECT MATH;
DISPLAY 1;ECOUNT:HYSTERESIS 1.000E+00;
POLARITY RISE;:MATH1:FILTER:DELAY:
TIME 0.000E+00;:MATH1:FILTER:IIR:
FORDER 2;HIPASS:COFF 10.00E+06;:MATH1:
FILTER:IIR:LOWPASS:COFF 10.00E+00;:
MATH1:FILTER:MAVG:WEIGHT 2;:MATH1:
FILTER:RESCALING:AVALUE 1.000E+00;
BVALUE 0.000E+00;:MATH1:FILTER:
TYPE THROUGH;:MATH1:INTEGRAL:PSCALING:
AVALUE 1.000E+00;BVALUE 0.000E+00;:
MATH1:INTEGRAL:RESCALING:
AVALUE 1.000E+00;BVALUE 0.000E+00;:
MATH1:INVERT 0;IPOINT:
POSITION 0.000E+00;:MATH1:LABEL:
DEFINE "MATH1";MODE 0;:MATH1:MINUS:
PSCALING1:AVALUE 1.000E+00;
BVALUE 0.000E+00;:MATH1:MINUS:
PSCALING2:AVALUE 1.000E+00;
BVALUE 0.000E+00;:MATH1:MINUS:
RESCALING:AVALUE 1.000E+00;
BVALUE 0.000E+00;:MATH1:MULTIPLE:
PSCALING1:AVALUE 1.000E+00;
BVALUE 0.000E+00;:MATH1:MULTIPLE:
PSCALING2:AVALUE 1.000E+00;
BVALUE 0.000E+00;:MATH1:MULTIPLE:
RESCALING:AVALUE 1.000E+00;
BVALUE 0.000E+00;:MATH1:
OPERATION FILTER,1;PLUS:PSCALING1:
AVALUE 1.000E+00;BVALUE ......
```

:MATH<x>:DISPlay

Function Turns ON/OFF the computed waveform 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>:ECOunt?

Function Queries all settings related to the edge count computation.

Syntax
```
:MATH<x>:ECOunt?
```
<x> = 1 to 8

Example
```
:MATH1:ECOUNT? ->
:MATH1:ECOUNT:HYSTERESIS 1.000E+00;
POLARITY RISE
```

:MATH<x>:ECOunt:HYSTeresis

Function Sets the hysteresis of the edge detection level of the edge count computation or queries the current setting.

Syntax
```
:MATH<x>:ECOunt:HYSTeresis {<NRf>}
:MATH<x>:ECOunt:HYSTeresis?
```
<x> = 1 to 8
<NRf> = 0 to 4 (div)

Example
```
:MATH1:ECOUNT:HYSTERESIS 1
:MATH1:ECOUNT:HYSTERESIS? ->
:MATH1:ECOUNT:HYSTERESIS 1.000E+00
```

:MATH<x>:ECOunt:POLarity

Function Sets the edge detection polarity of the edge count computation or queries the current setting.

Syntax
```
:MATH<x>:ECOunt:POLarity {FALL|RISE}
:MATH<x>:ECOunt:POLarity?
```
<x> = 1 to 8

Example
```
:MATH1:ECOUNT:POLARITY RISE
:MATH1:ECOUNT:POLARITY? ->
:MATH1:ECOUNT:POLARITY RISE
```

:MATH<x>:FILTer?

Function Queries all settings related to the filter.

Syntax
```
:MATH<x>:FILTer?
```
<x> = 1 to 8

Example
```
:MATH1:FILTER? -> :MATH1:FILTER:DELAY:
TIME 0.000E+00;:MATH1:FILTER:IIR:
FORDER 2;HIPASS:COFF 10.00E+06;:MATH1:
FILTER:IIR:LOWPASS:COFF 10.00E+06;:
MATH1:FILTER:MAVG:WEIGHT 2;:MATH1:
FILTER:RESCALING:AVALUE 1.000E+00;
BVALUE 0.000E+00;:MATH1:FILTER:
TYPE THROUGH
```

:MATH<x>:FILTer:DELay?

Function Queries all settings related to the delay computation.

Syntax
```
:MATH<x>:FILTer:DELay?
```
<x> = 1 to 8

Example
```
:MATH1:FILTER:DELAY? ->
:MATH1:FILTER:DELAY:TIME 0.000E+00
```

:MATH<x>:FILTer:DELay:TIME

Function Sets the delay value of the delay computation or queries the current setting.

Syntax
```
:MATH<x>:FILTer:DELay:TIME {<Time>}
:MATH<x>:FILTer:DELay:TIME?
```
<x> = 1 to 8
<Time> = See the DL9000 User's Manual.

Example
```
:MATH1:FILTER:DELAY:TIME 1S
:MATH1:FILTER:DELAY:TIME? ->
:MATH1:FILTER:DELAY:TIME 1.000E+00
```

:MATH<x>:FILTer:IIR?

Function Queries all settings related to the IIR filter computation.

Syntax
```
:MATH<x>:FILTer:IIR?
```
<x> = 1 to 8

Example
```
:MATH1:FILTER:IIR? -> :MATH1:FILTER:
IIR:FORDER 2;HIPASS:COFF 10.00E+06;:
MATH1:FILTER:IIR:LOWPASS:COFF 10.00E+00
```

:MATH<x>:FILTer:IIR:FORDer (Filter Order)

Function Sets the filter order of the IIR filter computation or queries the current setting.

Syntax
```
:MATH<x>:FILTer:IIR:FORDer {<NRf>}
:MATH<x>:FILTer:IIR:FORDer?
```
<x> = 1 to 8
<NRf> = 1 or 2

Example
```
:MATH1:FILTER:IIR:FORDER 2
:MATH1:FILTER:IIR:FORDER? ->
:MATH1:FILTER:IIR:FORDER 2
```

:MATH<x>:FILTer:IIR:HIPass?

Function Queries all settings related to the IIR high pass filter computation.

Syntax
```
:MATH<x>:FILTer:IIR:HIPass?
```
<x> = 1 to 8

Example
```
:MATH1:FILTER:IIR:HIPASS? ->
:MATH1:FILTER:IIR:HIPASS:COFF 10.00E+06
```

:MATH<x>:FILTer:IIR:HIPass:COFF

Function Sets the cutoff frequency of the IIR high pass filter computation or queries the current setting.

Syntax
```
:MATH<x>:FILTer:IIR:HIPass:
COFF {<Frequency>}
:MATH<x>:FILTer:IIR:HIPass:COFF?
```
<x> = 1 to 8
<Frequency> = 0.01 to 1 G (Hz)

Example
```
:MATH1:FILTER:IIR:HIPASS:COFF 10MHZ
:MATH1:FILTER:IIR:HIPASS:COFF? ->
:MATH1:FILTER:IIR:HIPASS:COFF 10.00E+06
```

:MATH<x>:FILTer:IIR:LOWPass?

Function Queries all settings related to the IIR low pass filter computation.

Syntax
```
:MATH<x>:FILTer:IIR:LOWPass?
```
<x> = 1 to 8

Example
```
:MATH1:FILTER:IIR:LOWPASS? ->
:MATH1:FILTER:IIR:LOWPASS:
COFF 10.00E+06
```

:MATH<x>:FILTer:IIR:LOWPass:COFF

Function Sets the cutoff frequency of the IIR low pass filter computation or queries the current setting.

Syntax
```
:MATH<x>:FILTer:IIR:LOWPass:C
OFF {<Frequency>}
:MATH<x>:FILTer:IIR:LOWPass:COFF?
```
<x> = 1 to 8
<Frequency> = 0.01 to 1 G (Hz)

Example
```
:MATH1:FILTER:IIR:LOWPASS:COFF 10HZ
:MATH1:FILTER:IIR:LOWPASS:COFF? ->
:MATH1:FILTER:IIR:LOWPASS:
COFF 10.00E+00
```

:MATH<x>:FILTer:MAVG?

Function Queries all settings related to the moving average computation.

Syntax
```
:MATH<x>:FILTer:MAVG?
```
<x> = 1 to 8

Example
```
:MATH1:FILTER:MAVG? ->
:MATH1:FILTER:MAVG:WEIGHT 2
```

:MATH<x>:FILTer:MAVG:WEIGht

Function Sets the weight of the moving average computation or queries the current setting.

Syntax
```
:MATH<x>:FILTer:MAVG:WEIGht {<NRf>}
:MATH<x>:FILTer:MAVG:WEIGht?
```
<x> = 1 to 8
<NRf> = 2 to 128 (2^n steps)

Example
```
:MATH1:FILTER:MAVG:WEIGHT 2
:MATH1:FILTER:MAVG:WEIGHT? ->
:MATH1:FILTER:MAVG:WEIGHT 2
```

:MATH<x>:FILTer:RESCaling?

Function Queries all settings related to the rescaling of the filter.

Syntax
```
:MATH<x>:FILTer:RESCaling?
```
<x> = 1 to 8

Example
```
:MATH1:FILTER:RESCALING? ->
:MATH1:FILTER:RESCALING:
AVALUE 1.000E+00;BVALUE 0.000E+00
```

:MATH<x>:FILTer:RESCaling:AVALue

Function Sets rescaling coefficient A of the filter or queries the current setting.

Syntax
```
:MATH<x>:FILTer:RESCaling:
AVALue {<NRf>}
:MATH<x>:FILTer:RESCaling:AVALue?
```
<x> = 1 to 8
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:FILTER:RESCALING:AVALUE 1
:MATH1:FILTER:RESCALING:AVALUE? ->
:MATH1:FILTER:RESCALING:
AVALUE 1.000E+00
```

:MATH<x>:FILTer:RESCaling:BVALue

Function Sets rescaling offset B of the filter or queries the current setting.

Syntax
```
:MATH<x>:FILTer:RESCaling:
BVALue {<NRf>}
:MATH<x>:FILTer:RESCaling:BVALue?
```
<x> = 1 to 8
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:FILTER:RESCALING:BVALUE 0
:MATH1:FILTER:RESCALING:BVALUE? ->
:MATH1:FILTER:RESCALING:
BVALUE 0.000E+00
```

:MATH<x>:FILTer:TYPE

Function Sets the filter type or queries the current setting.

Syntax
```
:MATH<x>:FILTer:TYPE {DELay|IHPass|
ILPass|MAVG|THRough}
```
:MATH<x>:FILTer:TYPE?
<x> = 1 to 8

Example
```
:MATH1:FILTER:TYPE DELAY
:MATH1:FILTER:TYPE? ->
:MATH1:FILTER:TYPE DELAY
```

:MATH<x>:INTegral?

Function Queries all settings related to the integral computation.

Syntax
```
:MATH<x>:INTegral?
```
<x> = 1 to 8

Example
```
:MATH1:INTEGRAL? -> :MATH1:INTEGRAL:
PSCALING:AVALUE 1.000E+00;
BVALUE 0.000E+00;:MATH1:INTEGRAL:
RESCALING:AVALUE 1.000E+00;
BVALUE 0.000E+00
```

:MATH<x>:INTegral:PSCaling?

Function Queries all settings related to the pre-scaling of the integral computation.

Syntax
```
:MATH<x>:INTegral:PSCaling?
```
<x> = 1 to 8

Example
```
:MATH1:INTEGRAL:PSCALING? ->
:MATH1:INTEGRAL:PSCALING:
AVALUE 1.000E+00;BVALUE 0.000E+00
```

:MATH<x>:INTegral:PSCaling:AVALue

Function Sets pre-scaling coefficient A of the integral computation or queries the current setting.

Syntax
```
:MATH<x>:INTegral:PSCaling:
AVALue {<NRf>}
:MATH<x>:INTegral:PSCaling:AVALue?
```
<x> = 1 to 8
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:INTEGRAL:PSCALING:AVALUE 1
:MATH1:INTEGRAL:PSCALING:AVALUE? ->
:MATH1:INTEGRAL:PSCALING:
AVALUE 1.000E+00
```

:MATH<x>:INTegral:PSCaling:BVALue

Function Sets pre-scaling offset B of the integral computation or queries the current setting.

Syntax
```
:MATH<x>:INTegral:PSCaling:
BVALue {<NRf>}
:MATH<x>:INTegral:PSCaling:BVALue?
```
<x> = 1 to 8
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:INTEGRAL:PSCALING:BVALUE 0
:MATH1:INTEGRAL:PSCALING:BVALUE? ->
:MATH1:INTEGRAL:PSCALING:
BVALUE 0.000E+00
```

:MATH<x>:INTegral:RESCaling?

Function Queries all settings related to the rescaling of the integral computation.

Syntax
```
:MATH<x>:INTegral:RESCaling?
```
<x> = 1 to 8

Example
```
:MATH1:INTEGRAL:RESCALING? ->
:MATH1:INTEGRAL:RESCALING:
AVALUE 1.000E+00;BVALUE 0.000E+00
```

:MATH<x>:INTegral:RESCaling:AVALue

Function Sets rescaling coefficient A of the integral computation or queries the current setting.

Syntax
```
:MATH<x>:INTegral:RESCaling:
AVALue {<NRf>}
:MATH<x>:INTegral:RESCaling:AVALue?
```
<x> = 1 to 8
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:INTEGRAL:RESCALING:AVALUE 1
:MATH1:INTEGRAL:RESCALING:AVALUE? ->
:MATH1:INTEGRAL:RESCALING:
AVALUE 1.000E+00
```

:MATH<x>:INTegral:RESCaling:BVALue

Function Sets rescaling offset B of the integral computation or queries the current setting.

Syntax
```
:MATH<x>:INTegral:RESCaling:
BVALue {<NRf>}
:MATH<x>:INTegral:RESCaling:BVALue?
```
<x> = 1 to 8
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:INTEGRAL:RESCALING:BVALUE 0
:MATH1:INTEGRAL:RESCALING:BVALUE? ->
:MATH1:INTEGRAL:RESCALING:
BVALUE 0.000E+00
```

:MATH<x>:INVert

Function Turns ON/OFF the inverted display of the computed waveform or queries the current setting.

Syntax
```
:MATH<x>:INVert {<Boolean>}
:MATH<x>:INVert?
```
<x> = 1 to 8

Example
```
:MATH1:INVERT ON
:MATH1:INVERT? -> :MATH1:INVERT 1
```

:MATH<x>:IPOint? (Initial Point)

Function Queries all settings related to the computation reference point.

Syntax
```
:MATH<x>:IPOint?
```
<x> = 1 to 8

Example
```
:MATH1:IPOINT? ->
:MATH1:IPOINT:POSITION 0.000E+00
```

:MATH<x>:IPOint:JUMP

Function Moves the computation reference point to the specified position.

Syntax
```
:MATH<x>:IPOint:JUMP {<NRf>|TRIGger|Z1|Z2}
```
<x> = 1 to 8
<NRf> = –5 or 0 (div)

Example
```
:MATH1:IPOINT:JUMP TRIGGER
```

:MATH<x>:IPOint:POSition

Function Sets the computation reference point or queries the current setting.

Syntax
```
:MATH<x>:IPOint:POSition {<NRf>}
:MATH<x>:IPOint:POSition?
```
<x> = 1 to 8
<NRf> = –5 to 5 (div)

Example
```
:MATH1:IPOINT:POSITION 0
:MATH1:IPOINT:POSITION? ->
:MATH1:IPOINT:POSITION 0.000E+00
```

Description This command is valid when `:MATH<x>:OPERation INTegral`.

:MATH<x>:LABel?

Function Queries all settings related to the label of the computed waveform.

Syntax
```
:MATH<x>:LABel?
```
<x> = 1 to 8

Example
```
:MATH1:LABEL? ->
:MATH1:LABEL:DEFINE "MATH1";MODE 0
```

:MATH<x>:LABel[:DEFine]

Function Sets the label of the computed waveform or queries the current setting.

Syntax
```
:MATH<x>:LABel[:DEFine] {<String>}
:MATH<x>:LABel[:DEFine]?
```
<x> = 1 to 8
<String> = Up to 8 characters

Example
```
:MATH1:LABEL:DEFINE "MATH1"
:MATH1:LABEL:DEFINE? ->
:MATH1:LABEL:DEFINE "MATH1"
```

:MATH<x>:LABel:MODE

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

Syntax
```
:MATH<x>:LABel:MODE {<Boolean>}
:MATH<x>:LABel:MODE?
```
<x> = 1 to 8

:MATH<x>:MINus?

Function Queries all settings related to the subtraction.

Syntax
```
:MATH<x>:MINus?
```
<x> = 1 to 8

Example
```
:MATH1:MINUS? -> :MATH1:MINUS:
PSCALING1:AVALUE 1.000E+00;
BVALUE 0.000E+00;:MATH1:MINUS:
PSCALING2:AVALUE 1.000E+00;
BVALUE 0.000E+00;:MATH1:MINUS:
RESCALING:AVALUE 1.000E+00;
BVALUE 0.000E+00+00
```

:MATH<x>:MINus:PSCaling<x>?

Function Queries all settings related to the pre-scaling of the subtraction.

Syntax
```
:MATH<x>:MINus:PSCaling<x>?
```
<x> of MATH<x> = 1 to 8
<x> of PSCaling<x> = 1 or 2

Example
```
:MATH1:MINUS:PSCALING1? ->
:MATH1:MINUS:PSCALING1:
AVALUE 1.000E+00;BVALUE 0.000E+00
```

:MATH<x>:MINus:PSCaling<x>:AVALue

Function Sets pre-scaling coefficient A of the subtraction or queries the current setting.

Syntax
```
:MATH<x>:MINus:PSCaling<x>:
AVALue {<NRf>}
:MATH<x>:MINus:PSCaling<x>:AVALue?
```
<x> of MATH<x> = 1 to 8
<x> of PSCaling<x> = 1 or 2
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:MINUS:PSCALING1:AVALUE 1
:MATH1:MINUS:PSCALING1:AVALUE? ->
:MATH1:MINUS:PSCALING1:AVALUE 1.000E+00
```

:MATH<x>:MINus:PSCaling<x>:BVALue

Function Sets pre-scaling offset B of the subtraction or queries the current setting.

Syntax
```
:MATH<x>:MINus:PSCaling<x>:
BVALue {<NRf>}
:MATH<x>:MINus:PSCaling<x>:BVALue?
```
<x> of MATH<x> = 1 to 8
<x> of PSCaling<x> = 1 or 2
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:MINUS:PSCALING1:BVALUE 0
:MATH1:MINUS:PSCALING1:BVALUE? ->
:MATH1:MINUS:PSCALING1:BVALUE 0.000E+00
```

:MATH<x>:MINus:RESCaling?

Function Queries all settings related to the rescaling of the subtraction.

Syntax
```
:MATH<x>:MINus:RESCaling?
```
<x> = 1 to 8

Example
```
:MATH1:MINUS:RESCALING? ->
:MATH1:MINUS:RESCALING:
AVALUE 1.000E+00;BVALUE 0.000E+00
```

:MATH<x>:MINus:RESCaling:AVALue

Function Sets rescaling coefficient A of the subtraction or queries the current setting.

Syntax
```
:MATH<x>:MINus:RESCaling:AVALue {<NRf>}
:MATH<x>:MINus:RESCaling:AVALue?
```
<x> = 1 to 8
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:MINUS:RESCALING:AVALUE 1
:MATH1:MINUS:RESCALING:AVALUE? ->
:MATH1:MINUS:RESCALING:AVALUE 1.000E+00
```

:MATH<x>:MINus:RESCaling:BVALue

Function Sets rescaling offset B of the subtraction or queries the current setting.

Syntax
```
:MATH<x>:MINus:RESCaling:BVALue {<NRf>}
:MATH<x>:MINus:RESCaling:BVALue?
```
<x> = 1 to 8
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:MINUS:RESCALING:BVALUE 0
:MATH1:MINUS:RESCALING:BVALUE? ->
:MATH1:MINUS:RESCALING:BVALUE 0.000E+00
```

:MATH<x>:MULTiple?

Function Queries all settings related to the multiplication.

Syntax
```
:MATH<x>:MULTiple?
```
<x> = 1 to 8

Example
```
:MATH1:MULTIPLE? -> :MATH1:MULTIPLE:
PSCALING1:AVALUE 1.000E+00;
BVALUE 0.000E+00;:MATH1:MULTIPLE:
PSCALING2:AVALUE 1.000E+00;
BVALUE 0.000E+00;:MATH1:MULTIPLE:
RESCALING:AVALUE 1.000E+00;
BVALUE 0.000E+00
```

:MATH<x>:MULTiple:PSCaling<x>?

Function Queries all settings related to the pre-scaling of the multiplication.

Syntax
```
:MATH<x>:MULTiple:PSCaling<x>?
```
<x> of MATH<x> = 1 to 8
<x> of PSCaling<x> = 1 or 2

Example
```
:MATH1:MULTIPLE:PSCALING1? ->
:MATH1:MULTIPLE:PSCALING1:
AVALUE 1.000E+00;BVALUE 0.000E+00
```

:MATH<x>:MULTiple:PSCaling<x>:AVALue

Function Sets pre-scaling coefficient A of the multiplication or queries the current setting.

Syntax
```
:MATH<x>:MULTiple:PSCaling<x>:
AVALue {<NRf>}
:MATH<x>:MULTiple:PSCaling<x>:AVALue?
```
<x> of MATH<x> = 1 to 8
<x> of PSCaling<x> = 1 or 2
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:MULTIPLE:PSCALING1:AVALUE 1
:MATH1:MULTIPLE:PSCALING1:AVALUE? ->
:MATH1:MULTIPLE:PSCALING1:
AVALUE 1.000E+00
```

:MATH<x>:MULTiple:PSCaling<x>:BVALue

Function Sets pre-scaling offset B of the multiplication or queries the current setting.

Syntax
```
:MATH<x>:MULTiple:PSCaling<x>:
BVALue {<NRf>}
:MATH<x>:MULTiple:PSCaling<x>:BVALue?
```
<x> of MATH<x> = 1 to 8
<x> of PSCaling<x> = 1 or 2
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:MULTIPLE:PSCALING1:BVALUE 0
:MATH1:MULTIPLE:PSCALING1:BVALUE? ->
:MATH1:MULTIPLE:PSCALING1:
BVALUE 0.000E+00
```

:MATH<x>:MULTiple:RESCaling?

Function Queries all settings related to the rescaling of the multiplication.

Syntax
```
:MATH<x>:MULTiple:RESCaling?
```
<x> = 1 to 8

Example
```
:MATH1:MULTIPLE:RESCALING? ->
:MATH1:MULTIPLE:RESCALING:
AVALUE 1.000E+00;BVALUE 0.000E+00
```

:MATH<x>:MULTiple:RESCaling:AVALue

Function Sets rescaling coefficient A of the multiplication or queries the current setting.

Syntax
```
:MATH<x>:MULTiple:RESCaling:
AVALue {<NRf>}
:MATH<x>:MULTiple:RESCaling:AVALue?
```
<x> = 1 to 8
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:MULTIPLE:RESCALING:AVALUE 1
:MATH1:MULTIPLE:RESCALING:AVALUE? ->
:MATH1:MULTIPLE:RESCALING:
AVALUE 1.000E+00
```

:MATH<x>:MULTiple:RESCaling:BVALue

Function Sets rescaling offset B of the multiplication or queries the current setting.

Syntax
```
:MATH<x>:MULTiple:RESCaling:
BVALue {<NRf>}
:MATH<x>:MULTiple:RESCaling:BVALue?
```
<x> = 1 to 8
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:MULTIPLE:RESCALING:BVALUE 0
:MATH1:MULTIPLE:RESCALING:BVALUE? ->
:MATH1:MULTIPLE:RESCALING:
BVALUE 0.000E+00
```

:MATH<x>:OPERation

Function Sets the operator or queries the current setting.

Syntax
```
:MATH<x>:OPERation {(ECOunt|FILTer|
INTegral|MINus|MULTiple|PLUS|RCOunt),
<NRf>,<NRf>}
:MATH<x>:OPERation?
```
<x> = 1 to 8
<NRf> = 1 to 4

Example
```
:MATH1:OPERATION FILTER,1
:MATH1:OPERATION? ->
:MATH1:OPERATION FILTER,1
```

Description For unary operators (ECOunt|FILTer|INTegral), select the target waveform using the first <NRf>.
For binary operators (MINus|MULTiple|PLUS|RCOunt), select the target waveform of the first term using the first <NRf> and the target waveform of the second term using the second <NRf>.

:MATH<x>:PLUS?

Function Queries all settings related to the addition.

Syntax
```
:MATH<x>:PLUS?
```
<x> = 1 to 8

Example
```
:MATH1:PLUS? -> :MATH1:PLUS:
PSCALING1:AVALUE 1.000E+00;
BVALUE 0.000E+00;:MATH1:PLUS:PSCALING2:
AVALUE 1.000E+00;BVALUE 0.000E+00;:
MATH1:PLUS:RESCALING:AVALUE 1.000E+00;
BVALUE 0.000E+00
```

:MATH<x>:PLUS:PSCaling<x>?

Function Queries all settings related to the pre-scaling of the addition.

Syntax
```
:MATH<x>:PLUS:PSCaling<x>?
```
<x> of MATH<x> = 1 to 8
<x> of PSCaling<x> = 1 or 2

Example
```
:MATH1:PLUS:PSCALING1? ->
:MATH1:PLUS:PSCALING1:
AVALUE 1.000E+00;BVALUE 0.000E+00
```

:MATH<x>:PLUS:PSCaling<x>:AVALue

Function Sets pre-scaling coefficient A of the addition or queries the current setting.

Syntax
```
:MATH<x>:PLUS:PSCaling<x>:
AVALue {<NRf>}
:MATH<x>:PLUS:PSCaling<x>:AVALue?
```
<x> of MATH<x> = 1 to 8
<x> of PSCaling<x> = 1 or 2
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:PLUS:PSCALING1:AVALUE 1
:MATH1:PLUS:PSCALING1:AVALUE? ->
:MATH1:PLUS:PSCALING1:AVALUE 1.000E+00
```

:MATH<x>:PLUS:PSCaling<x>:BVALue

Function Sets pre-scaling offset B of the addition or queries the current setting.

Syntax
```
:MATH<x>:PLUS:PSCaling<x>:
BVALue {<NRf>}
:MATH<x>:PLUS:PSCaling<x>:BVALue?
```
<x> of MATH<x> = 1 to 8
<x> of PSCaling<x> = 1 or 2
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:PLUS:PSCALING1:BVALUE 0
:MATH1:PLUS:PSCALING1:BVALUE? ->
:MATH1:PLUS:PSCALING1:BVALUE 0.000E+00
```

:MATH<x>:PLUS:RESCaling?

Function Queries all settings related to the rescaling of the addition.

Syntax
```
:MATH<x>:PLUS:RESCaling?
```
<x> = 1 to 8

Example
```
:MATH1:PLUS:RESCALING? ->
:MATH1:PLUS:RESCALING:AVALUE 1.000E+00;
BVALUE 0.000E+00
```

:MATH<x>:PLUS:RESCaling:AVALue

Function Sets rescaling coefficient A of the addition or queries the current setting.

Syntax
```
:MATH<x>:PLUS:RESCaling:AVALue {<NRf>}
:MATH<x>:PLUS:RESCaling:AVALue?
```
<x> = 1 to 8
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:PLUS:RESCALING:AVALUE 1
:MATH1:PLUS:RESCALING:AVALUE? ->
:MATH1:PLUS:RESCALING:AVALUE 1.000E+00
```

:MATH<x>:PLUS:RESCaling:BVALue

Function Sets rescaling offset B of the addition or queries the current setting.

Syntax
```
:MATH<x>:PLUS:RESCaling:BVALue {<NRf>}
:MATH<x>:PLUS:RESCaling:BVALue?
```
<x> = 1 to 8
<NRf> = –1.0000E+31 to 1.0000E+31

Example
```
:MATH1:PLUS:RESCALING:BVALUE 0
:MATH1:PLUS:RESCALING:BVALUE? ->
:MATH1:PLUS:RESCALING:BVALUE 0.000E+00
```

:MATH<x>:POSition

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

Syntax
```
:MATH<x>:POSition {<NRf>}
:MATH<x>:POSition?
```
<x> = 1 to 8
<NRf> = –4 to 4 (div)

Example
```
:MATH1:POSITION 0
:MATH1:POSITION? ->
:MATH1:POSITION 0.000E+00
```

:MATH<x>:SCALe?

Function Queries all settings related to scaling.

Syntax
```
:MATH<x>:SCALe?
```
<x> = 1 to 8

Example
```
:MATH1:SCALE? -> :MATH1:SCALE:
CENTER 1.000E+00;MODE AUTO;
SENSITIVITY 1.000E+00
```

:MATH<x>:SCALe:CENTer

Function Sets the offset of the computed waveform or queries the current setting.

Syntax
```
:MATH<x>:SCALe:CENTer {<NRf>|<Voltage>|
<Current>}
:MATH<x>:SCALe:CENTer?
```
<x> = 1 to 8
<NRf>, <Voltage>, and <Current> = See the DL9000 User's Manual.

Example
```
:MATH1:SCALE:CENTER 1
:MATH1:SCALE:CENTER? ->
:MATH1:SCALE:CENTER 1.000E+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:SENSitivity

Function Sets the vertical sensitivity of the computed waveform or queries the current setting.

Syntax
```
:MATH<x>:SCALe:SENSitivity {<NRf>|
<Voltage>|<Current>}
:MATH<x>:SCALe:SENSitivity?
```
<x> = 1 to 8
<NRf>, <Voltage>, and <Current> = See the DL9000 User's Manual.

Example
```
:MATH1:SCALE:SENSITIVITY 1
:MATH1:SCALE:SENSITIVITY? ->
:MATH1:SCALE:SENSITIVITY 1.000E+00
```

:MATH<x>:SELect

Function Sets the display option or queries the current setting.

Syntax
```
:MATH<x>:SELect {INPut|MATH|REFerence}
:MATH<x>:SELect?
```
<x> = 1 to 8

Example
```
:MATH1:SELECT INPUT
:MATH1:SELECT? -> :MATH1:SELECT INPUT
```

:MATH<x>:SVALue (Scale VALUE)

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

Syntax
```
:MATH<x>:SVALue {<Boolean>}
:MATH<x>:SVALue?
```
<x> = 1 to 8

Example
```
:MATH1:SVALUE ON
:MATH1:SVALUE? -> MATH1:SVALUE 1
```

:MATH<x>:THReshold<x>

Function Sets the edge detection level of the count computation or queries the current setting.

Syntax
```
:MATH<x>:THReshold<x> {<NRf>|
<Voltage>|<Current>}
:MATH<x>:THReshold<x>?
```
<x> of MATH<x> = 1 to 8
<x> of THReshold<x> = 1 or 2
<NRf>, <Voltage>, and <Current> = See the DL9000 User's Manual.

Example
```
:MATH1:THRESHOLD1 1
:MATH1:THRESHOLD1? ->
:MATH1:THRESHOLD1 1.000E+00
```

Description THReshold2 is valid when the operation is `{RCOunt}`.

:MATH<x>:UNIT?

Function Queries all settings related to the computation unit.

Syntax
```
:MATH<x>:UNIT?
```
<x> = 1 to 8

Example
```
:MATH1:UNIT? ->
:MATH1:UNIT:DEFINE "EU";MODE AUTO
```

:MATH<x>:UNIT[:DEFine]

Function Sets the computation unit or queries the current setting.

Syntax
```
:MATH<x>:UNIT[:DEFine] {<String>}
:MATH<x>:UNIT[:DEFine]?
```
<x> = 1 to 8
<String> = Up to 4 characters

Example
```
:MATH1:UNIT:DEFINE "EU"
:MATH1:UNIT:DEFINE? ->
:MATH1:UNIT:DEFINE "EU"
```

:MATH<x>:UNIT:MODE

Function Sets the automatic/manual addition of the computation unit or queries the current setting.

Syntax
```
:MATH<x>:UNIT:MODE {AUTO|USERdefine}
:MATH<x>:UNIT:MODE?
```
<x> = 1 to 8

Example
```
:MATH1:UNIT:MODE AUTO
:MATH1:UNIT:MODE? ->
:MATH1:UNIT:MODE AUTO
```

5.18 MEASure Group

:MEASure?

Function Queries all settings related to the automated measurement of waveform parameters.

Syntax :MEASure?

Example :MEASURE? ->
:MEASURE:CALCULATION:DEFINE1 “MAX(C1)”;
DEFINE2 “MIN(C2)”;DEFINE3 “HIGH(C3)”;
DEFINE4 “LOW(C4)”;STATE1 0;STATE2 0;
STATE3 0;STATE4 0;:MEASURE:CONTINUOUS:
COUNT 0;:MEASURE:CYCLE:TRACE 1;:
MEASURE:DISPLAY 1;MODE BASIC;THRESHOLD:
TRACE1:AUTO PTOPEAK;LHYSTERESIS:
HYSTERESIS 1.000E+00;LEVEL 0.000E+00;:
MEASURE:THRESHOLD:TRACE1:MODE AUTO;
ULOWER:RANGE 2.000E+00,1.000E+00;:
MEASURE:THRESHOLD:TRACE2:AUTO PTOPEAK;
LHYSTERESIS:HYSTERESIS 1.000E+00;
LEVEL 0.000E+00;:MEASURE:THRESHOLD:
TRACE2:MODE AUTO;ULOWER:
RANGE 2.000E+00,1.000E+00;:MEASURE:
THRESHOLD:TRACE3:AUTO PTOPEAK;
LHYSTERESIS:HYSTERESIS 1.000E+00;
LEVEL 0.000E+00;:MEASURE:THRESHOLD:
TRACE3:MODE AUTO;ULOWER:
RANGE 2.000E+00,1.000E+00;:MEASURE:
THRESHOLD:TRACE4:AUTO PTOPEAK;
LHYSTERESIS:HYSTERESIS 1.000E+00;
LEVEL 0.000E+00;:MEASURE:THRESHOLD:
TRACE4:MODE AUTO;ULOWER:
RANGE 2.000E+00,1.000E+00;:MEASURE:
THRESHOLD:TRACE5:AUTO PTOPEAK;
LHYSTERESIS:HYSTERESIS 1.000E+00;
LEVEL 0.000E+00;:MEASURE:THRESHOLD:
TRACE5:MODE AUTO;ULOWER:RANGE ......

:MEASure:CALCulation?

Function Queries all settings related to calculation items.

Syntax :MEASure:CALCulation?

Example :MEASURE:CALCULATION? ->
:MEASURE:CALCULATION:DEFINE1 “MAX(C1)”;
DEFINE2 “MIN(C2)”;DEFINE3 “HIGH(C3)”;
DEFINE4 “LOW(C4)”;STATE1 0;STATE2 0;
STATE3 0;STATE4 0

:MEASure:CALCulation:ALL

Function Turns ON/OFF all calculation items.

Syntax :MEASure:CALCulation:ALL {<Boolean>}

Example :MEASURE:CALCULATION:ALL ON

:MEASure:CALCulation:COUNt<x>?

Function Queries the statistical processing count of the calculation item.

Syntax :MEASure:CALCulation:COUNt<x>?
<x> = 1 to 4

Example :MEASURE:CALCULATION:COUNT1? ->
:MEASURE:CALCULATION:COUNT1 1

:MEASure:CALCulation:DEFine<x>

Function Sets the equation of the calculation item or queries the current setting.

Syntax :MEASure:CALCulation:DEFine<x>
{<String>}
:MEASure:CALCulation:DEFine<x>?
<x> = 1 to 4
<String> = Up to 128 characters

Example :MEASURE:CALCULATION:DEFINE1 “MAX(C1)”
:MEASURE:CALCULATION:DEFINE1? ->
:MEASURE:CALCULATION:DEFINE1 “MAX(C1)”

:MEASure:CALCulation:{MAXimum<x>|MEAN<x>|MINimum<x>|SDEViation<x>}?

Function Queries the statistical value of the calculation item.

Syntax :MEASure:CALCulation:{MAXimum<x>|MEAN<x>|MINimum<x>|SDEViation<x>}?
<x> of MAXimum<x> = 1 to 4
<x> of MEAN<x> = 1 to 4
<x> of MINimum<x> = 1 to 4
<x> of SDEViation<x> = 1 to 4

Example (The following is an example for the maximum value.)
:MEASURE:CALCULATION:MAXIMUM1? ->
:MEASURE:CALCULATION:MAXIMUM1 1.000E+00

:MEASure:CALCulation:STATe<x>

Function Turns ON/OFF the calculation item or queries the current setting.

Syntax :MEASure:CALCulation:STATe<x>
{<Boolean>}
:MEASure:CALCulation:STATe<x>?
<x> = 1 to 4

Example :MEASURE:CALCULATION:STATE1 ON
:MEASURE:CALCULATION:STATE1? ->
:MEASURE:CALCULATION:STATE1 1

:MEASure:CALCulation:VALue<x>?

Function Queries the automated measured value of the calculation item.

Syntax `:MEASure:CALCulation:VALue<x>? {<NRf>}`
<x> = 1 to 4
<NRf> = 1 to 100000

Example
```
:MEASURE:CALCULATION:VALUE1?  ->
:MEASURE:CALCULATION:VALUE1 1.000E+00
```

Description
- If the measurement is not possible, "NAN (Not A Number" is returned.
- <NRf> indicates the nth automated measured value in the past.
  In the case of cycle statistical processing, specify the <NRf>th cycle from the left of the screen.
  To specify the oldest automated measured value, specify 1.
  If <NRf> is omitted, the latest automated measured value is specified.
  If the value corresponding to the relevant count is not present, "NAN" (Not A Number)" is returned.

:MEASure:CONTinuous?

Function Queries all settings related to the continuous statistical processing.

Syntax `:MEASure:CONTinuous?`

Example
```
:MEASURE:CONTINUOUS? ->
:MEASURE:CONTINUOUS:COUNT 0
```

:MEASure:CONTinuous:COUNt

Function Sets the continuous statistical processing count or queries the current setting.

Syntax
```
:MEASure:CONTinuous:COUNt {<NRf>}
:MEASure:CONTinuous:COUNt?
```
<NRf> = 0 to 100000

Example
```
:MEASURE:CONTINUOUS:COUNT 10
:MEASURE:CONTINUOUS:COUNT? ->
:MEASURE:CONTINUOUS:COUNT 10
```

Description When <NRf> = 0, the maximum count that is possible under the current settings is automatically set.

:MEASure:CONTinuous:RESTart

Function Restarts the continuous statistical processing.

Syntax `:MEASure:CONTinuous:RESTart`

Example `:MEASURE:CONTINUOUS:RESTART`

Description Clears the previous statistical data.

:MEASure:CYCLe?

Function Queries all settings related to the cycle statistical processing.

Syntax `:MEASure:CYCLe?`

Example
```
:MEASURE:CYCLE? ->
:MEASURE:CYCLE:TRACE 1
```

:MEASure:CYCLe:ABORt

Function Aborts the execution of the cycle statistical processing.

Syntax `:MEASure:CYCLe:ABORt`

Example `:MEASURE:CYCLE:ABORT`

:MEASure:CYCLe:EXECute

Function Executes the cycle statistical processing. This is an overlap command.

Syntax `:MEASure:CYCLe:EXECute`

Example `:MEASURE:CYCLE:EXECUTE`

Description Continues the operation without clearing the previous statistical data.

:MEASure:CYCLe:TRACe

Function Sets the cycle source trace of the continuous statistical processing count or queries the current setting.

Syntax
```
:MEASure:CYCLe:TRACe {<NRf>}
:MEASure:CYCLe:TRACe?
```
<NRf> = 1 to 8

Example
```
:MEASURE:CYCLE:TRACE 1
:MEASURE:CYCLE:TRACE? ->
:MEASURE:CYCLE:TRACE 1
```

:MEASure:DISPlay

Function Turns ON/OFF the display of the automated measurement of waveform parameters or queries the current setting.

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

Example
```
:MEASURE:DISPLAY ON
:MEASURE:DISPLAY? -> :MEASURE:DISPLAY 1
```

:MEASure:HISTory:ABORt

Function Aborts the execution of the statistical processing of the history data.

Syntax `:MEASure:HISTory:ABORt`

Example `:MEASURE:HISTORY:ABORT`

:MEASure:HISTory:EXECute

Function Executes the statistical processing of the history data. This is an overlap command.

Syntax `:MEASure:HISTory:EXECute`

Example `:MEASURE:HISTORY:EXECUTE`

:MEASure:MODE

Function Sets the mode of the automated measurement of waveform parameters or queries the current setting.

Syntax
```
:MEASure:MODE {BASic|CONTinuous|CYCLe|
HISTory}
:MEASure:MODE?
```

Example
```
:MEASURE:MODE BASIC
:MEASURE:MODE? -> :MEASURE:MODE BASIC
```

:MEASure:THReshold?

Function Queries all settings related to the threshold levels of the automated measurement of waveform parameters.

Syntax :MEASure:THReshold?

Example
```
:MEASURE:THRESHOLD? ->
:MEASURE:THRESHOLD:TRACE1:AUTO PTOPEAK;
LHYSTERESIS:HYSTERESIS 1.000E+00;
LEVEL 0.000E+00;:MEASURE:THRESHOLD:
TRACE1:MODE AUTO;ULOWER:
RANGE 2.000E+00,1.000E+00;:MEASURE:
THRESHOLD:TRACE2:AUTO PTOPEAK;
LHYSTERESIS:HYSTERESIS 1.000E+00;
LEVEL 0.000E+00;:MEASURE:THRESHOLD:
TRACE2:MODE AUTO;ULOWER:
RANGE 2.000E+00,1.000E+00;:MEASURE:
THRESHOLD:TRACE3:AUTO PTOPEAK;
LHYSTERESIS:HYSTERESIS 1.000E+00;
LEVEL 0.000E+00;:MEASURE:THRESHOLD:
TRACE3:MODE AUTO;ULOWER:
RANGE 2.000E+00,1.000E+00;:MEASURE:
THRESHOLD:TRACE4:AUTO PTOPEAK;
LHYSTERESIS:HYSTERESIS 1.000E+00;
LEVEL 0.000E+00;:MEASURE:THRESHOLD:
TRACE4:MODE AUTO;ULOWER:
RANGE 2.000E+00,1.000E+00;:MEASURE:
THRESHOLD:TRACE5:AUTO PTOPEAK;
LHYSTERESIS:HYSTERESIS 1.000E+00;
LEVEL 0.000E+00;:MEASURE:THRESHOLD:
TRACE5:MODE AUTO;ULOWER:
RANGE 2.000E+00,1.000E+00;:MEASURE:
THRESHOLD:TRACE6:AUTO PTOPEAK;
LHYSTERESIS:HYSTERESIS 1.000E+00;
LEVEL 0.000E+00;:MEASURE:THRESHOLD:
TRACE6:MODE AUTO;ULOWER:RANGE ......
```

Description This command is valid when the <Parameter> of :MEASure:TRACe<x>:AREA<x>:<Parameter> is not RISE or FALL.

:MEASure:THReshold:TRACe<x>?

Function Queries the threshold levels of the trace.

Syntax :MEASure:THReshold:TRACe<x>?
<x> = 1 to 8

Example
```
:MEASURE:THRESHOLD:TRACE1? ->
:MEASURE:THRESHOLD:TRACE1:AUTO PTOPEAK;
LHYSTERESIS:HYSTERESIS 1.000E+00;
LEVEL 0.000E+00;:MEASURE:THRESHOLD:
TRACE1:MODE AUTO;ULOWER:
RANGE 1.000E+00,2.000E+00
```

Description This command is valid when the <Parameter> of :MEASure:TRACe<x>:AREA<x>:<Parameter> is not RISE or FALL.

:MEASure:THReshold:TRACe<x>:AUTO

Function Sets the detection mode when the auto setting of the threshold level is enabled or queries the current setting.

Syntax :MEASure:THReshold:TRACe<x>:AUTO {HLOW|PTOPeak}
:MEASure:THReshold:TRACe<x>:AUTO?
<x> = 1 to 8

Example
```
:MEASURE:THRESHOLD:TRACE1:AUTO PTOPEAK
:MEASURE:THRESHOLD:TRACE1:AUTO? ->
:MEASURE:THRESHOLD:TRACE1:AUTO PTOPEAK
```

Description This command is valid when the <Parameter> of :MEASure:TRACe<x>:AREA<x>:<Parameter> is not RISE or FALL.

:MEASure:THReshold:TRACe<x>:LHYSteresis?

Function Queries all settings related to the level and hysteresis of the threshold level.

Syntax :MEASure:THReshold:TRACe<x>:LHYSteresis?
<x> = 1 to 8

Example
```
:MEASURE:THRESHOLD:TRACE1:LHYSTERESIS?
-> :MEASURE:THRESHOLD:TRACE1:
LHYSTERESIS:HYSTERESIS 1.000E+00;
LEVEL 0.000E+00
```

Description This command is valid when the <Parameter> of :MEASure:TRACe<x>:AREA<x>:<Parameter> is not RISE or FALL.

:MEASure:THReshold:TRACe<x>:LHYSteresis:HYSTeresis

Function Sets the hysteresis of the threshold level or queries the current setting.

Syntax :MEASure:THReshold:TRACe<x>:LHYSteresis:HYSTeresis {<NRf>}
:MEASure:THReshold:TRACe<x>:LHYSteresis:HYSTeresis?
<x> = 1 to 8
<NRf> = 0 to 4 (div)

Example
```
:MEASURE:THRESHOLD:TRACE1:LHYSTERESIS:
HYSTERESIS 1
:MEASURE:THRESHOLD:TRACE1:LHYSTERESIS:
HYSTERESIS? ->
:MEASURE:THRESHOLD:TRACE1:
LHYSTERESIS:HYSTERESIS 1.000E+00
```

Description This command is valid when the <Parameter> of :MEASure:TRACe<x>:AREA<x>:<Parameter> is not RISE or FALL.

:MEASure:THReshold:TRACe<x>:LHYSteresis:LEVel

Function Sets the level of the threshold level or queries the current setting.

Syntax
```
:MEASure:THReshold:TRACe<x>:
LHYSteresis:LEVel
{<NRf>|<Voltage>|<Current>}
:MEASure:THReshold:TRACe<x>:
LHYSteresis:LEVel?
```
<x> = 1 to 8
<NRf>, <Voltage>, and <Current> = See the DL9000 User's Manual.

Example
```
:MEASURE:THRESHOLD:TRACE1:LHYSTERESIS:
LEVEL 1
:MEASURE:THRESHOLD:TRACE1:LHYSTERESIS:
LEVEL? -> :MEASURE:THRESHOLD:TRACE1:
LHYSTERESIS:LEVEL 1.000E+00
```

Description This command is valid when the <Parameter> of :MEASure:TRACe<x>:AREA<x>:<Parameter> is not RISE or FALL.

:MEASure:THReshold:TRACe<x>:MODE

Function Sets the setup mode of the threshold level or queries the current setting.

Syntax
```
:MEASure:THReshold:TRACe<x>:MODE {AUTO|
LHYSteresis|ULOWer}
:MEASure:THReshold:TRACe<x>:MODE?
```
<x> = 1 to 8

Example
```
:MEASURE:THRESHOLD:TRACE1:
MODE LHYSTERESIS
:MEASURE:THRESHOLD:TRACE1:MODE? ->
:MEASURE:THRESHOLD:TRACE1:
MODE LHYSTERESIS
```

Description This command is valid when the <Parameter> of :MEASure:TRACe<x>:AREA<x>:<Parameter> is not RISE or FALL.

:MEASure:THReshold:TRACe<x>:ULOWer?

Function Queries all settings related to the upper and lower limits of the threshold level.

Syntax
```
:MEASure:THReshold:TRACe<x>:ULOWer?
```
<x> = 1 to 8

Example
```
:MEASURE:THRESHOLD:TRACE1:ULOWER? ->
:MEASURE:THRESHOLD:TRACE1:ULOWER:
RANGE 2.000E+00,1.000E+00
```

Description This command is valid when the <Parameter> of :MEASure:TRACe<x>:AREA<x>:<Parameter> is not RISE or FALL.

:MEASure:THReshold:TRACe<x>:ULOWer:RANGe

Function Sets the upper and lower limits of the threshold level or queries the current setting.

Syntax
```
:MEASure:THReshold:TRACe<x>:ULOWer:
RANGe {(<NRf>,<NRf>)|
(<Voltage>,<Voltage>)|
(<Current>,<Current>)}
:MEASure:THReshold:TRACe<x>:ULOWer:
RANGe?
```
<x> = 1 to 8
<NRf>, <Voltage>, and <Current> = See the DL9000 User's Manual.

Example
```
:MEASURE:THRESHOLD:TRACE1:ULOWER:
RANGE 1,2
:MEASURE:THRESHOLD:TRACE1:ULOWER:RANGE?
-> :MEASURE:THRESHOLD:TRACE1:ULOWER:
RANGE 2.000E+00,1.000E+00
```

Description This command is valid when the <Parameter> of :MEASure:TRACe<x>:AREA<x>:<Parameter> is not RISE or FALL.

:MEASure:TRACe<x>?

Function Queries all settings related to the trace.

Syntax
```
:MEASure:TRACe<x>?
```
<x> = 1 to 8

Example
```
:MEASURE:TRACE1? ->
:MEASURE:TRACE1:AREA1:BURST:STATE 1;:
MEASURE:TRACE1:AREA1:CMEAN:STATE 0;:
MEASURE:TRACE1:AREA1:COUNT:STATE 0;:
MEASURE:TRACE1:AREA1:CRMS:STATE 0;:
MEASURE:TRACE1:AREA1:CSDEVIATION:
STATE 0;:MEASURE:TRACE1:AREA1:DELAY:
MEASURE:COUNT 3;POLARITY RISE;:MEASURE:
TRACE1:AREA1:DELAY:REFERENCE:COUNT 1;
POLARITY FALL;TRACE 2;:MEASURE:TRACE1:
AREA1:DELAY:SOURCE TRACE;:MEASURE:
TRACE1:AREA1:DPROXIMAL:MODE PERCENT;
PERCENT 10,90;UNIT -1.000E+00,
1.000E+00;:MEASURE:TRACE1:AREA1:
DUTYCYCLE:STATE 1;:MEASURE:TRACE1:
AREA1:FALL:STATE 0;:MEASURE:TRACE1:
AREA1:FREQUENCY:STATE 0;:MEASURE:
TRACE1:AREA1:HIGH:STATE 1;:MEASURE:
TRACE1:AREA1:HILOW:STATE 0;:MEASURE:
TRACE1:AREA1:LOW:STATE 1;:MEASURE:
TRACE1:AREA1:MAXIMUM:STATE 1;:MEASURE:
TRACE1:AREA1:MEAN:STATE 1;:MEASURE:
TRACE1:AREA1:MINIMUM:STATE 1;:MEASURE:
TRACE1:AREA1:NOVERSHOOT:STATE 0;:
MEASURE:TRACE1:AREA1:NWIDTH:STATE 1;:
MEASURE:TRACE1:AREA1:PERFREQUENCY:
STATE 0;:MEASURE:TRACE1:AREA1:PERIOD:
STATE 0;:MEASURE:TRACE1:AREA1:
POVERSHOOT:STATE ......
```

:MEASure:TRACe<x>:AREA<x>?

Function Queries all settings related to the area.

Syntax :MEASure:TRACe<x>:AREA<x>?
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2

Example
```
:MEASURE:TRACE1:AREA1? ->
:MEASURE:TRACE1:AREA1:BURST:STATE 1;:
MEASURE:TRACE1:AREA1:CMEAN:STATE 0;:
MEASURE:TRACE1:AREA1:COUNT:STATE 0;:
MEASURE:TRACE1:AREA1:CRMS:STATE 0;:
MEASURE:TRACE1:AREA1:CSDEVIATION:
STATE 0;:MEASURE:TRACE1:AREA1:DELAY:
MEASURE:COUNT 3;POLARITY RISE;:MEASURE:
TRACE1:AREA1:DELAY:REFERENCE:COUNT 1;
POLARITY FALL;TRACE 2;:MEASURE:TRACE1:
AREA1:DELAY:SOURCE TRACE;:MEASURE:
TRACE1:AREA1:DPROXIMAL:MODE PERCENT;
PERCENT 10,90;UNIT -1.000E+00,
1.000E+00;:MEASURE:TRACE1:AREA1:
DUTYCYCLE:STATE 1;:MEASURE:TRACE1:
AREA1:FALL:STATE 0;:MEASURE:TRACE1:
AREA1:FREQUENCY:STATE 0;:MEASURE:
TRACE1:AREA1:HIGH:STATE 1;:MEASURE:
TRACE1:AREA1:HILOW:STATE 0;:MEASURE:
TRACE1:AREA1:LOW:STATE 1;:MEASURE:
TRACE1:AREA1:MAXIMUM:STATE 1;:MEASURE:
TRACE1:AREA1:MEAN:STATE 1;:MEASURE:
TRACE1:AREA1:MINIMUM:STATE 1;:MEASURE:
TRACE1:AREA1:NOVERSHOOT:STATE 0;:
MEASURE:TRACE1:AREA1:NWIDTH:STATE 1;:
MEASURE:TRACE1:AREA1:PERFREQUENCY:
STATE 0;:MEASURE:TRACE1:AREA1:PERIOD:
STATE 0;:MEASURE:TRACE1:AREA1:
POVERSHOOT:STATE ......
```

:MEASure:TRACe<x>:AREA<x>:ALL

Function Turns ON/OFF all waveform parameters.

Syntax :MEASure:TRACe<x>:AREA<x>:ALL
{<Boolean>}
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2

Example
```
:MEASURE:TRACE1:AREA1:ALL ON
```

:MEASure:TRACe<x>:AREA<x>:<Parameter>?

Function Queries all settings related to the waveform parameter.

Syntax :MEASure:TRACe<x>:AREA<x>:<Parameter>?
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2
<Parameter> = {BURSt|CMEan|COUNt|CRMS|CSDeviation|DELay|DUTYcycle|FALL|FREQuency|HIGH|HILow|LOW|MAXimum|MEAN|MINimum|NOVershoot|NWIDth|PERFrequency|PERiod|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEViation|TYCInteg|TYINteg}

Example (The following is an example for the maximum value of trace 1 and area 1.)
```
:MEASURE:TRACE1:AREA1:MAXIMUM? ->
:MEASURE:TRACE1:AREA1:MAXIMUM:STATE 0
```

:MEASure:TRACe<x>:AREA<x>:<Parameter>:COUNt?

Function Queries the continuous statistical processing count of the waveform parameter.

Syntax :MEASure:TRACe<x>:AREA<x>:<Parameter>:COUNt?
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2
<Parameter> = {BURSt|CMEan|COUNt|CRMS|CSDeviation|DELay|DUTYcycle|FALL|FREQuency|HIGH|HILow|LOW|MAXimum|MEAN|MINimum|NOVershoot|NWIDth|PERFrequency|PERiod|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEViation|TYCInteg|TYINteg}

Example (The following is an example for the maximum value of trace 1 and area 1.)
```
:MEASURE:TRACE1:AREA1:MAXIMUM:COUNT? ->
:MEASURE:TRACE1:AREA1:MAXIMUM:COUNT 0
```

:MEASure:TRACe<x>:AREA<x>:<Parameter>:{MAXimum|MEAN|MINimum|SDEViation}?

Function Queries the statistical value of the waveform parameter.

Syntax
```
:MEASure:TRACe<x>:AREA<x>:<Parameter>:
{MAXimum|MEAN|MINimum|SDEViation}?
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2
<Parameter> = {BURSt|CMEan|COUNt|CRMS|CSDeviation|DELay|DUTYcycle|FALL|FREQuency|HIGH|HILow|LOW|MAXimum|MEAN|MINimum|NOVershoot|NWIDth|PERFrequency|PERiod|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEViation|TYCInteg|TYINteg}

Example (The following is an example for the maximum value.)
```
:MEASURE:TRACE1:AREA1:MAXIMUM:
MAXIMUM? ->
:MEASURE:TRACE1:AREA1:MAXIMUM:
MAXIMUM 1.000E+00
```

:MEASure:TRACe<x>:AREA<x>:<Parameter>:STATe

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

Syntax
```
:MEASure:TRACe<x>:AREA<x>:<Parameter>:
STATe {<Boolean>}
:MEASure:TRACe<x>:AREA<x>:<Parameter>:
STATe?
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2
<Parameter> = {BURSt|CMEan|COUNt|CRMS|CSDeviation|DELay|DUTYcycle|FALL|FREQuency|HIGH|HILow|LOW|MAXimum|MEAN|MINimum|NOVershoot|NWIDth|PERFrequency|PERiod|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEViation|TYCInteg|TYINteg}

Example (The following is an example for the maximum value of trace 1 and area 1.)
```
:MEASURE:TRACE1:AREA1:MAXIMUM:STATE ON
:MEASURE:TRACE1:AREA1:MAXIMUM:STATE? ->
:MEASURE:TRACE1:AREA1:MAXIMUM:STATE 1
```

:MEASure:TRACe<x>:AREA<x>:<Parameter>:VALue?

Function Queries the automated measured value of the waveform parameter.

Syntax
```
:MEASure:TRACe<x>:AREA<x>:<Parameter>:
VALue? {<NRf>}
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2
<Parameter> = {BURSt|CMEan|COUNt|CRMS|CSDeviation|DELay|DUTYcycle|FALL|FREQuency|HIGH|HILow|LOW|MAXimum|MEAN|MINimum|NOVershoot|NWIDth|PERFrequency|PERiod|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEViation|TYCInteg|TYINteg}
<NRf> = 1 to 100000

Example (The following is an example for the maximum value of trace 1 and area 1.)
```
:MEASURE:TRACE1:AREA1:MAXIMUM:VALUE? ->
:MEASURE:TRACE1:AREA1:MAXIMUM:
VALUE 1.000E+00
```

Description
- If the measurement is not possible, “NAN (Not A Number” is returned.
- <NRf> indicates the nth automated measured value in the past.
  In the case of cycle statistical processing, specify the <NRf> cycle from the left of the screen.
  To specify the oldest automated measured value, specify 1.
  If <NRf> is omitted, the latest automated measured value is specified.
  If the value corresponding to the relevant count is not present, “NAN” (Not A Number)” is returned.

:MEASure:TRACe<x>:AREA<x>:DELay:MEASure?

Function Queries all settings related to the measurement conditions of the source waveform of the delay measurement between channels.

Syntax
```
:MEASure:TRACe<x>:AREA<x>:DELay:
MEASure?
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2

Example
```
:MEASURE:TRACE1:AREA1:DELAY:MEASURE? ->
:MEASURE:TRACE1:AREA1:DELAY:MEASURE:
COUNT 1;POLARITY RISE
```

:MEASure:TRACe<x>:AREA<x>:DELay:MEASure:COUNt

Function Sets the edge detection count of the source waveform of the delay measurement between channels or queries the current setting.

Syntax
```
:MEASure:TRACe<x>:AREA<x>:DELay:MEASure:COUNt {<NRf>}
:MEASure:TRACe<x>:AREA<x>:DELay:MEASure:COUNt?
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2
<NRf> = 1 to 10

Example
```
:MEASURE:TRACE1:AREA1:DELAY:MEASURE:COUNT 1
:MEASURE:TRACE1:AREA1:DELAY:MEASURE:COUNT? -> :MEASURE:TRACE1:AREA1:DELAY:MEASURE:COUNT 1
```

:MEASure:TRACe<x>:AREA<x>:DELay:MEASure:POLarity

Function Sets the polarity of the source waveform of the delay measurement between channels or queries the current setting.

Syntax
```
:MEASure:TRACe<x>:AREA<x>:DELay:MEASure:POLarity {FALL|RISE}
:MEASure:TRACe<x>:AREA<x>:DELay:MEASure:POLarity?
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2

Example
```
:MEASURE:TRACE1:AREA1:DELAY:MEASURE:POLARITY RISE
:MEASURE:TRACE1:AREA1:DELAY:MEASURE:POLARITY? -> :MEASURE:TRACE1:AREA1:DELAY:MEASURE:POLARITY RISE
```

:MEASure:TRACe<x>:AREA<x>:DELay:REFerence?

Function Queries all settings related to the reference waveform of the delay measurement between channels.

Syntax
```
:MEASure:TRACe<x>:AREA<x>:DELay:REFerence?
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2

Example
```
:MEASURE:TRACE1:AREA1:DELAY:REFERENCE? -> :MEASURE:TRACE1:AREA1:DELAY:REFERENCE:COUNT 1;POLARITY FALL;TRACE 1
```

:MEASure:TRACe<x>:AREA<x>:DELay:REFerence:COUNt

Function Sets the edge detection count of the reference waveform of the delay measurement between channels or queries the current setting.

Syntax
```
:MEASure:TRACe<x>:AREA<x>:DELay:REFerence:COUNt {<NRf>}
:MEASure:TRACe<x>:AREA<x>:DELay:REFerence:COUNt?
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2
<NRf> = 1 to 10

Example
```
:MEASURE:TRACE1:AREA1:DELAY:REFERENCE:COUNT 1
:MEASURE:TRACE1:AREA1:DELAY:REFERENCE:COUNT? -> :MEASURE:TRACE1:AREA1:DELAY:REFERENCE:COUNT 1
```

:MEASure:TRACe<x>:AREA<x>:DELay:REFerence:POLarity

Function Sets the polarity of the reference waveform of the delay measurement between channels or queries the current setting.

Syntax
```
:MEASure:TRACe<x>:AREA<x>:DELay:REFerence:POLarity {FALL|RISE}
:MEASure:TRACe<x>:AREA<x>:DELay:REFerence:POLarity?
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2

Example
```
:MEASURE:TRACE1:AREA1:DELAY:REFERENCE:POLARITY FALL
:MEASURE:TRACE1:AREA1:DELAY:REFERENCE:POLARITY? -> :MEASURE:TRACE1:AREA1:DELAY:REFERENCE:POLARITY FALL
```

:MEASure:TRACe<x>:AREA<x>:DELay:REFerence:TRACe

Function Sets the trace of the reference waveform of the delay measurement between channels or queries the current setting.

Syntax
```
:MEASure:TRACe<x>:AREA<x>:DELay:REFerence:TRACe {<NRf>}
:MEASure:TRACe<x>:AREA<x>:DELay:REFerence:TRACe?
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2
<NRf> = 1 to 8

Example
```
:MEASURE:TRACE1:AREA1:DELAY:REFERENCE:TRACE 1
:MEASURE:TRACE1:AREA1:DELAY:REFERENCE:TRACE? -> :MEASURE:TRACE1:AREA1:DELAY:REFERENCE:TRACE 1
```

:MEASure:TRACe<x>:AREA<x>:DELay:SOURce

Function Sets the reference of the delay measurement between channels or queries the current setting.

Syntax
```
:MEASure:TRACe<x>:AREA<x>:DELay:SOURce {TRACe|TRIGger}
:MEASure:TRACe<x>:AREA<x>:DELay:SOURce?
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2

Example
```
:MEASURE:TRACE1:AREA1:DELAY:SOURCE TRACE
:MEASURE:TRACE1:AREA1:DELAY:SOURCE? ->
:MEASURE:TRACE1:AREA1:DELAY:SOURCE TRACE
```

:MEASure:TRACe<x>:AREA<x>:DPRoximal?

Function Queries all settings related to the distal and proximal values.

Syntax
```
:MEASure:TRACe<x>:AREA<x>:DPRoximal?
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2

Example
```
:MEASURE:TRACE1:AREA1:DPROXIMAL? ->
;MEASURE:TRACE1:AREA1:DPROXIMAL:MODE PERCENT;PERCENT 10,20;UNIT 1.000E+00,1.000E+00
```

Description This command is valid when the <Parameter> of :MEASure:TRACe<x>:AREA<x>:<Parameter> is RISE or FALL.

:MEASure:TRACe<x>:AREA<x>:DPRoximal:MODE

Function Sets the unit of the distal and proximal values or queries the current setting.

Syntax
```
:MEASure:TRACe<x>:AREA<x>:DPRoximal:MODE {PERCent|UNIT}
:MEASure:TRACe<x>:AREA<x>:DPRoximal:MODE?
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2

Example
```
:MEASURE:TRACE1:AREA1:DPROXIMAL:MODE PERCENT
:MEASURE:TRACE1:AREA1:DPROXIMAL:MODE? -> :MEASURE:TRACE1:AREA1:DPROXIMAL:MODE PERCENT
```

Description This command is valid when the <Parameter> of :MEASure:TRACe<x>:AREA<x>:<Parameter> is RISE or FALL.

:MEASure:TRACe<x>:AREA<x>:DPRoximal:PERCent

Function Sets the distal and proximal values as a percentage or queries the current setting.

Syntax
```
:MEASure:TRACe<x>:AREA<x>:DPRoximal:PERCent {<NRf>,<NRf>}
:MEASure:TRACe<x>:AREA<x>:DPRoximal:PERCent?
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2
<NRf> = 0 to 100 (%)

Example
```
:MEASURE:TRACE1:AREA1:DPROXIMAL:PERCENT 10,90
:MEASURE:TRACE1:AREA1:DPROXIMAL:PERCENT? -> :MEASURE:TRACE1:AREA1:DPROXIMAL:PERCENT 10,90
```

Description This command is valid when the <Parameter> of :MEASure:TRACe<x>:AREA<x>:<Parameter> is RISE or FALL.

:MEASure:TRACe<x>:AREA<x>:DPRoximal:UNIT

Function Sets the distal and proximal values in the specified unit or queries the current setting.

Syntax
```
:MEASure:TRACe<x>:AREA<x>:DPRoximal:UNIT {(<NRf>,<NRf>)|(<Voltage>,<Voltage>)|(<Current>,<Current>)}
:MEASure:TRACe<x>:AREA<x>:DPRoximal:UNIT?
```
<x> of TRACe<x> = 1 to 8
<x> of AREA<x> = 1 or 2
<NRf>, <Voltage>, and <Current> = See the DL9000 User's Manual.

Example
```
:MEASURE:TRACE1:AREA1:DPROXIMAL:UNIT 1,-1
:MEASURE:TRACE1:AREA1:DPROXIMAL:UNIT? -> :MEASURE:TRACE1:AREA1:DPROXIMAL:UNIT -1.000E+00,1.000E+00
```

Description This command is valid when the <Parameter> of :MEASure:TRACe<x>:AREA<x>:<Parameter> is RISE or FALL.

:MEASure:TRANge<x> (Time Range)

Function Sets the measurement range or queries the current setting.

Syntax
```
:MEASure:TRANge<x> {<NRf>,<NRf>}
:MEASure:TRANge<x>?
```
<x> = 1 or 2
<NRf> = –5 to 5 (div)

Example
```
:MEASURE:TRANGE1 -5,0
:MEASURE:TRANGE1? ->
:MEASURE:TRANGE1 0.000E+00,-5.000E+00
```

:MEASure:WAIT?

Function Waits for the completion of the automated measurement with a timeout option.

Syntax `MEASure:WAIT? {<NRf>}`
<NRf> = 1 to 360000 (timeout value, in units of 10 ms)

Example `MEASURE:WAIT 100? -> :MEASURE:WAIT 1`

Description If the execution of the automated measurement completes within the timeout value, 0 is returned; if it is not complete or automated measurement is not being executed, 1 is returned.
Even if the timeout value is set long, 0 is returned when the execution of the automated measurement is complete.

:MEASure:WINDow<x>

Function Sets the measurement source window of the area or queries the current setting.

Syntax `:MEASure:WINDow<x> {MAIN|Z1|Z2}`
`:MEASure:WINDow<x>?`
<x> = 1 or 2

Example `:MEASURE:WINDOW1 MAIN`
`:MEASURE:WINDOW1? ->`
`:MEASURE:WINDOW1 MAIN`

5.19 REFerence Group

:REFerence<x>?

Function Queries all settings related to the reference.
Syntax :REFerence<x>?
<x> = 1 to 4
Example :REFERENCE1? -> :REFERENCE1:
SELECT REFERENCE;DISPLAY 1;INVERT 0;
LABEL:DEFINE “REF1”;MODE 1;:REFERENCE1:
POSITION 1.000E+00;SVALUE 1

:REFerence<x>:DISPlay

Function Turns ON/OFF the display of the reference or queries the current setting.
Syntax :REFerence<x>:DISPlay {<Boolean>}
:REFerence<x>:DISPlay?
<x> = 1 to 4
Example :REFERENCE1:DISPLAY ON
:REFERENCE1:DISPLAY? ->
:REFERENCE1:DISPLAY 1

:REFerence<x>:INVert

Function Sets the inverted display of the reference or queries the current setting.
Syntax :REFerence<x>:INVert {<Boolean>}
:REFerence<x>:INVert?
<x> = 1 to 4
Example :REFERENCE1:INVERT ON
:REFERENCE1:INVERT? ->
:REFERENCE1:INVERT 1

:REFerence<x>:LABel?

Function Queries all settings related to the waveform label of the reference.
Syntax :REFerence<x>:LABel?
<x> = 1 to 4
Example :REFERENCE1:LABEL? ->
:REFERENCE1:LABEL:DEFINE “REF1”;MODE 1

:REFerence<x>:LABel[:DEFine]

Function Sets the waveform label of the reference or queries the current setting.
Syntax :REFerence<x>:LABel[:DEFine] {<String>}
:REFerence<x>:LABel[:DEFine]?
<x> = 1 to 4
<String> = Up to 8 characters
Example :REFERENCE1:LABEL:DEFINE “REF1”
:REFERENCE1:LABEL:DEFINE? ->
:REFERENCE1:LABEL:DEFINE “REF1”

:REFerence<x>:LABel:MODE

Function Turns ON/OFF the waveform label display of the reference or queries the current setting.
Syntax :REFerence<x>:LABel:MODE {<Boolean>}
:REFerence<x>:LABel:MODE?
<x> = 1 to 4
Example :REFERENCE1:LABEL:MODE ON
:REFERENCE1:LABEL:MODE? ->
:REFERENCE1:LABEL:MODE 1

:REFerence<x>:LOAD

Function Loads the waveform to the reference.
Syntax :REFerence<x>:LOAD {<NRf>}
<x> = 1 to 4
<NRf> = 1 to12 (1 to 8 are traces and 9 to 12 are internal memories)
Example :REFERENCE1:LOAD 1

:REFerence<x>:POSition

Function Sets the vertical position of the reference or queries the current setting.
Syntax :REFerence<x>:POSition {<NRf>}
:REFerence<x>:POSition?
<x> = 1 to 4
<NRf> = –4 to 4 (div)
Example :REFERENCE1:POSITION 1
:REFERENCE1:POSITION? ->
:REFERENCE1:POSITION 1.000E+00

:REFerence<x>:SELect

Function Sets the waveform (computation or reference) to the computation channel or queries the current setting.
Syntax :REFerence<x>:SELect {MATH|REFerence}
:REFerence<x>:SELect?
<x> = 1 to 4
Example :REFERENCE1:SELECT MATH
:REFERENCE1:SELECT? ->
:REFERENCE1:SELECT MATH

:REFerence<x>:SVALue (Scale VALUE)

Function Turns ON/OFF the scale display of the reference or queries the current setting.
Syntax :REFerence<x>:SVALue {<Boolean>}
:REFerence<x>:SVALue?
<x> = 1 to 4
Example :REFERENCE1:SVALUE ON
:REFERENCE1:SVALUE? ->
:REFERENCE1:SVALUE 1

5.20 SEARch Group

:SEARch<x>?

Function Queries all settings related to the search function.

Syntax :SEARch<x>?

<x> = 1 or 2

Example :SEARCH1? -> :SEARCH1:CLOCK:SOURCE 1;
POLARITY FALL;:SEARCH1:DECIMATION 1;
HOLDOFF 0.000E+00;LOGIC AND;
POLARITY RISE;SMODE OFF;SPATTERN:CLOCK:
MODE 1;POLARITY FALL;SOURCE 1;:SEARCH1:
SPATTERN:CS 1;LATCH:TRACE 1;
POLARITY FALL;:SEARCH1:SPATTERN:SETUP:
BITRATE 1.000E+00;
PATTERN "1100110111101111";:SEARCH1:
SPOINT -5.000E+00;STRACE 1;TRACE1:
CONDITION DONTCARE;
HYSTERESIS 1.000E+00;LEVEL 0.000E+00;:
SEARCH1:TRACE2:CONDITION DONTCARE;
HYSTERESIS 1.000E+00;LEVEL 0.000E+00;:
SEARCH1:TRACE3:CONDITION DONTCARE;
HYSTERESIS 1.000E+00;LEVEL 0.000E+00;:
SEARCH1:TRACE4:CONDITION DONTCARE;
HYSTERESIS 1.000E+00;LEVEL 0.000E+00;:
SEARCH1:TRACE5:CONDITION DONTCARE;
HYSTERESIS 1.000E+00;LEVEL 0.000E+00;:
SEARCH1:TRACE6:CONDITION DONTCARE;
HYSTERESIS 1.000E+00;LEVEL 0.000E+00;:
SEARCH1:TRACE7:CONDITION DONTCARE;
HYSTERESIS 1.000E+00;LEVEL 0.000E+00;:
SEARCH1:TRACE8:CONDITION DONTCARE;
HYSTERESIS 1.000E+00;LEVEL 0.000E+00;:
SEARCH1:TYPE EDGE;WIDTH:MODE OUT;
TIME1 1.000E+00;TIME2 2.000E+00;
TYPE PULSE

:SEARch<x>:ABORt

Function Aborts the search.

Syntax :SEARch<x>:ABORt

<x> = 1 or 2

Example :SEARCH1:ABORT

:SEARch<x>:CLOCk?

Function Queries all settings related to the clock channel.

Syntax :SEARch<x>:CLOCk?

<x> = 1 or 2

Example :SEARCH1:CLOCK? ->
:SEARCH1:CLOCK:SOURCE 1;POLARITY RISE

:SEARch<x>:CLOCk:POLarity

Function Sets the polarity of the clock channel or queries the current setting.

Syntax :SEARch<x>:CLOCk:POLarity {FALL|RISE}
:SEARch<x>:CLOCk:POLarity?

<x> = 1 or 2

Example :SEARCH1:CLOCK:POLARITY RISE
:SEARCH1:CLOCK:POLARITY? ->
:SEARCH1:CLOCK:POLARITY RISE

Description
- This command is invalid when :SEARch<x>:CLOCk:SOURce NONE.
- This command is valid when :SEARch<x>:TYPE STATe.
- This command is valid when :SEARch<x>:TYPE WIDTh and :SEARch<x>:WIDTh:TYPE PSTAte.

:SEARch<x>:CLOCk:SOURce

Function Sets the clock trace of the search or queries the current setting.

Syntax :SEARch<x>:CLOCk:SOURce {<NRf>|NONE}
:SEARch<x>:CLOCk:SOURce?

<x> = 1 or 2

<NRf> = 1 to 8

Example :SEARCH1:CLOCK:SOURCE NONE
:SEARCH1:CLOCK:SOURCE? ->
:SEARCH1:CLOCK:SOURCE NONE

Description
- This command is valid when :SEARch<x>:TYPE STATe.
- This command is valid when :SEARch<x>:TYPE WIDTh and :SEARch<x>:WIDTh:TYPE PSTAte.

:SEARch<x>:DECimation

Function Sets the decimation detection of the skip mode or queries the current setting.

Syntax :SEARch<x>:DECimation {<NRf>}
:SEARch<x>:DECimation?

<x> = 1 or 2

<NRf> = 1 to 9999

Example :SEARCH1:DECIMATION 1
:SEARCH1:DECIMATION? ->
:SEARCH1:DECIMATION 1

:SEARch<x>:EXECute

Function Executes the search. This is an overlap command.

Syntax
```
:SEARch<x>:EXECute
```
<x> = 1 or 2

Example
```
:SEARCH1:EXECUTE
```

:SEARch<x>:HOLDoff

Function Sets the hold off detection or queries the current setting.

Syntax
```
:SEARch<x>:HOLDoff {<Time>}
:SEARch<x>:HOLDoff?
```
<x> = 1 or 2
<Time> = 0 s to 1 s (100 ps steps)

Example
```
:SEARCH1:HOLDOFF 0S
:SEARCH1:HOLDOFF? ->
:SEARCH1:HOLDOFF 0.000E+00
```

:SEARch<x>:LOGic

Function Sets the search logic or queries the current setting.

Syntax
```
:SEARch<x>:LOGic {AND|OR}
:SEARch<x>:LOGic?
```
<x> = 1 or 2

Example
```
:SEARCH1:LOGIC OR
:SEARCH1:LOGIC? -> :SEARCH1:LOGIC OR
```

Description
- This command is valid when `:SEARch<x>:TYPE EQUalify|SPATtern|STATe`.
- This command is valid when `:SEARch<x>:TYPE WIDTh` and `:SEARch<x>:WIDTh:TYPE PQUalify|PSTAte`.

:SEARch<x>:POLarity

Function Sets the search polarity or queries the current setting.

Syntax
```
:SEARch<x>:POLarity {ENTer|EXIT|FALL|FALSe|NEGative|POSitive|RISE|TRUE}
:SEARch<x>:POLarity?
```
<x> = 1 or 2

Example
```
:SEARCH1:POLARITY ENTER
:SEARCH1:POLARITY? ->
:SEARCH1:POLARITY ENTER
```

Description
- `{FALL|RISE}` is valid when `:SEARch<x>:TYPE EDGE|EQUalify`.
- `{ENTer|EXIT}` is valid when `:SEARch<x>:TYPE STATe`.
- `{NEGative|POSitive}` is valid when `:SEARch<x>:TYPE WIDTh` and `:SEARch<x>:WIDTh:TYPE PQUalify|PULSe`.
- `{FALSe|TRUE}` is valid when `:SEARch<x>:TYPE WIDTh` and `:SEARch<x>:WIDTh:TYPE PSTAte`.

:SEARch<x>:SELect

Function Sets the detection waveform number of the search function and queries the position that corresponds to the detection waveform number.

Syntax
```
:SEARch<x>:SELect {<NRf>|MAXimum}
:SEARch<x>:SELect?
```
<x> = 1 or 2
<NRf> = 0 to 4999

Example
```
:SEARCH1:SELECT 1
:SEARCH1:SELECT? ->
:SEARCH1:SELECT 1.500E+00
```

Description If there is no searched position, “NAN” is returned for the query.

:SEARch<x>:SELect? MAXimum

Function Queries the detection count of the search function.

Syntax
```
:SEARch<x>:SELect? {MAXimum}
```
<x> = 1 or 2

Example
```
:SEARCH1:SEECT? MAXIMUM ->
:SEARCH1:SELECT 100
```

Description If there is no searched position, “NAN” is returned for the query.

:SEARch<x>:SMODe

Function Sets the skip mode or queries the current setting.

Syntax
```
:SEARch<x>:SMODe {DECimation|HOLDoff|OFF}
:SEARch<x>:SMODe?
```
<x> = 1 or 2

Example
```
:SEARCH1:SMODE HOLDOFF
:SEARCH1:SMODE? ->
:SEARCH1:SMODE HOLDOFF
```

:SEARch<x>:SPATtern? (Serial Pattern)

Function Queries all settings related to the serial pattern search.

Syntax
```
:SEARch<x>:SPATtern?
```
<x> = 1,2

Example
```
:SEARCH1:SPATTERN? -> :SEARCH1:SPATTERN:CLOCK:MODE 1;POLARITY FALL;SOURCE 1;:SEARCH1:SPATTERN:CS 1;LATCH:TRACE 1;POLARITY FALL;:SEARCH1:SPATTERN:SETUP:BITRATE 1.000E+00;PATTERN "1100110111101111"
```

:SEARch<x>:SPATtern:CLOCk?

Function Queries all settings related to clock of the serial pattern search.

Syntax
```
:SEARch<x>:SPATtern:CLOCk?
```
<x> = 1,2

Example
```
:SEARCH1:SPATTERN:CLOCK? ->
:SEARCH1:SPATTERN:CLOCK:MODE 1;POLARITY FALL;SOURCE 1
```

:SEARch<x>:SPATtern:CLOCk:MODE

Function Enables/Disables the clock of the serial pattern search or queries the current setting.

Syntax
```
:SEARch<x>:SPATtern:CLOCk:
MODE {<Boolean>}
:SEARch<x>:SPATtern:CLOCk:MODE?
```
<x> = 1,2

Example
```
:SEARCH1:SPATTERN:CLOCK:MODE ON
:SEARCH1:SPATTERN:CLOCK:MODE? ->
:SEARCH1:SPATTERN:CLOCK:MODE 1
```

:SEARch<x>:SPATtern:CLOCk:POLarity

Function Sets the polarity of the clock trace of the serial pattern search or queries the current setting.

Syntax
```
:SEARch<x>:SPATtern:CLOCk:
POLarity {FALL|RISE}
:SEARch<x>:SPATtern:CLOCk:POLarity?
```
<x> = 1,2

Example
```
:SEARCH1:SPATTERN:CLOCK:POLARITY FALL
:SEARCH1:SPATTERN:CLOCK:POLARITY? ->
:SEARCH1:SPATTERN:CLOCK:POLARITY FALL
```

Description This command is valid when
```
:SEARch<x>:SPATtern:CLOCk:MODE ON.
```

:SEARch<x>:SPATtern:CLOCk:SOURce

Function Sets the clock trace of the serial pattern search or queries the current setting.

Syntax
```
:SEARch<x>:SPATtern:CLOCk:
SOURce {<NRf>}
:SEARch<x>:SPATtern:CLOCk:SOURce?
```
<x> = 1,2
<NRf> = 1 to 8

Example
```
:SEARCH1:SPATTERN:CLOCK:SOURCE 1
:SEARCH1:SPATTERN:CLOCK:SOURCE? ->
:SEARCH1:SPATTERN:CLOCK:SOURCE 1
```

Description This command is valid when
```
:SEARch<x>:SPATtern:CLOCk:MODE ON.
```

:SEARch<x>:SPATtern:CS

Function Enables/Disables the chip select of the serial pattern search or queries the current setting.

Syntax
```
:SEARch<x>:SPATtern:CS {<Boolean>}
:SEARch<x>:SPATtern:CS?
```
<x> = 1,2

Example
```
:SEARCH1:SPATTERN:CS ON
:SEARCH1:SPATTERN:CS? ->
:SEARCH1:SPATTERN:CS 1
```

Description This command is valid when
```
:SEARch<x>:SPATtern:CLOCk:MODE ON.
POLARITY FALL
```

:SEARch<x>:SPATtern:LATCh?

Function Queries all settings related to latch of the serial pattern search.

Syntax
```
:SEARch<x>:SPATtern:LATCh?
```
<x> = 1,2

Example
```
:SEARCH1:SPATTERN:LATCH? ->
:SEARCH1:SPATTERN:LATCH:TRACE 1;
POLARITY FALL
```

:SEARch<x>:SPATtern:LATCh:POLarity

Function Sets the polarity of the latch trace of the serial pattern search or queries the current setting.

Syntax
```
:SEARch<x>:SPATtern:LATCh:
POLarity {FALL|RISE}
:SEARch<x>:SPATtern:LATCh:POLarity?
```
<x> = 1,2

Example
```
:SEARCH1:SPATTERN:LATCH:POLARITY FALL
:SEARCH1:SPATTERN:LATCH:POLARITY? ->
:SEARCH1:SPATTERN:LATCH:POLARITY FALL
```

Description
- This command is valid when
  `:SEARch<x>:SPATtern:CLOCk:MODE ON.`
- This command is invalid when
  `:SEARch<x>:SPATtern:LATCh:TRACe NONE.`

:SEARch<x>:SPATtern:LATCh:TRACe

Function Sets the latch trace of the serial pattern search or queries the current setting.

Syntax
```
:SEARch<x>:SPATtern:LATCh:TRACe {<NRf>|
NONE}
:SEARch<x>:SPATtern:LATCh:TRACe?
```
<x> = 1,2
<NRf> = 1 to 8

Example
```
:SEARCH1:SPATTERN:LATCH:TRACE 1
:SEARCH1:SPATTERN:LATCH:TRACE? ->
:SEARCH1:SPATTERN:LATCH:TRACE 1
```

Description This command is valid when
```
:SEARch<x>:SPATtern:CLOCk:MODE ON.
```

:SEARch<x>:SPATtern[:SETup]?

Function Queries all settings related to setup of the serial pattern search.

Syntax
```
:SEARch<x>:SPATtern[:SETup]?
```
<x> = 1,2

Example
```
:SEARCH1:SPATTERN:SETUP? ->
:SEARCH1:SPATTERN:SETUP:
BITRATE 1.000E+03;DATA:ACTIVE HIGH;
TRACE 1;
:SEARCH1:SPATTERN:SETUP:PATTERN "1100"
```

:SEARch<x>:SPATtern[:SETup]:BITRate

Function Sets the bit rate of the serial pattern search or queries the current setting.

Syntax
```
:SEARch<x>:SPATtern[:SETup]:
BITRate {<NRf>}
:SEARch<x>:SPATtern[:SETup]:BITRate?
```
<x> = 1, 2
<NRf> = 1 to 1G (bps)

Example
```
:SEARCH1:SPATTERN:SETUP:BITRATE 1
:SEARCH1:SPATTERN:SETUP:BITRATE? ->
:SEARCH1:SPATTERN:SETUP:
BITRATE 1.000E+00
```

Description This command is valid when `:SEARch<x>:SPATtern:CLOCk:MODE OFF`.

:SEARch<x>:SPATtern[:SETup]:CLEar

Function Clears the entire pattern of the serial pattern search (to don't care).

Syntax
```
:SEARch<x>:SPATtern[:SETup]:CLEar
```
<x> = 1,2

Example
```
:SEARCH1:SPATTERN:SETUP:CLEAR
```

:SEARch<x>:SPATtern[:SETup]:DATA?

Function Queries all settings related to data of the serial pattern search.

Syntax
```
:SEARch<x>:SPATtern[:SETup]:DATA?
```
<x> = 1,2

Example
```
:SEARCH1:SPATTERN:SETUP:DATA? -
>:SEARCH1:SPATTERN:SETUP:DATA:
ACTIVE HIGH;TRACE 1
```

:SEARch<x>:SPATtern[:SETup]:DATA:ACTive

Function Sets the active level of the data trace of the serial pattern search or queries the current setting.

Syntax
```
:SEARch<x>:SPATtern[:SETup]:DATA:
ACTive {HIGH|LOW}
:SEARch<x>:SPATtern[:SETup]:DATA:
ACTive?
```
<x> = 1,2

Example
```
:SEARCH1:SPATTERN:SETUP:DATA:
ACTIVE HIGH
:SEARCH1:SPATTERN:SETUP:DATA:ACTIVE? ->
:SEARCH1:SPATTERN:SETUP:DATA:
ACTIVE HIGH
```

:SEARch<x>:SPATtern[:SETup]:DATA:TRACe

Function Sets the data trace of the serial pattern search or queries the current setting.

Syntax
```
:SEARch<x>:SPATtern[:SETup]:DATA:
TRACe {<NRf>}
:SEARch<x>:SPATtern[:SETup]:DATA:TRACe?
```
<x> = 1,2
<NRf> = 1 to 8

Example
```
:SEARCH1:SPATTERN:SETUP:DATA:TRACE 1
:SEARCH1:SPATTERN:SETUP:DATA:TRACE? ->
:SEARCH1:SPATTERN:SETUP:DATA:TRACE 1
```

:SEARch<x>:SPATtern[:SETup]:HEXA

Function Sets the pattern of the serial pattern search in hexadecimal notation.

Syntax
```
:SEARch<x>:SPATtern[:SETup]:HEXA
{<String>}
```
<x> = 1,2
<String> = Up to 32 characters by combining '0' to 'F' and 'X'

Example
```
:SEARCH1:SPATTERN:SETUP:HEXA "ABCD"
```

:SEARch<x>:SPATtern[:SETup]:PATTern

Function Sets the pattern of the serial pattern search in binary notation or queries the current setting.

Syntax
```
:SEARch<x>:SPATtern[:SETup]:
PATTern {<String>}
:SEARch<x>:SPATtern[:SETup]:PATTern?
```
<x> = 1,2
<String> = Up to 128 characters by combining '0', '1', and 'X'

Example
```
:SEARCH1:SPATTERN:SETUP:
PATTERN "1100110111101111"
:SEARCH1:SPATTERN:SETUP:PATTERN? ->
:SEARCH1:SPATTERN:SETUP:
PATTERN "1100110111101111"
```

:SEARch<x>:SPOint

Function Sets the search start position or queries the current setting.

Syntax
```
:SEARch<x>:SPOint {<NRf>}
:SEARch<x>:SPOint?
```
<x> = 1 or 2
<NRf> = –5 to 5 (div)

Example
```
:SEARCH1:SPOINT 1
:SEARCH1:SPOINT? ->
:SEARCH1:SPOINT 1.000E+00
```

:SEARch<x>:STRace

Function Sets the search source trace or queries the current setting.

Syntax
```
:SEARch<x>:STRace {<NRf>}
:SEARch<x>:STRace?
```
<x> = 1 or 2
<NRf> = 1 to 8

Example
```
:SEARCH1:STRACE 1
:SEARCH1:STRACE? -> :SEARCH1:STRACE 1
```

Description
- This command is valid when `:SEARch<x>:TYPE EDGE|EQUalify`.
- This command is valid when `:SEARch<x>:TYPE WIDTh` and `:SEARch<x>:WIDTh:TYPE PQUalify|PULSe`.

:SEARch<x>:TRACe<x>?

Function Queries all settings related to the search conditions of the trace.

Syntax
```
:SEARch<x>:TRACe<x>?
```
<x> of SEARch<x> = 1 or 2
<x> of TRACe<x> = 1 to 8

Example
```
:SEARCH1:TRACE1? ->
:SEARCH1:TRACE1:CONDITION DONTCARE;
HYSTERESIS 1.000E+00;LEVEL 0.000E+00
```

:SEARch<x>:TRACe<x>:CONDition

Function Sets the condition to be satisfied for the trace or queries the current setting.

Syntax
```
:SEARch<x>:TRACe<x>:
CONDition {DONTcare|HIGH|LOW}
:SEARch<x>:TRACe<x>:CONDition?
```
<x> of SEARch<x> = 1 or 2
<x> of TRACe<x> = 1 to 8

Example
```
:SEARCH1:TRACE1:CONDITION HIGH
:SEARCH1:TRACE1:CONDITION? ->
:SEARCH1:TRACE1:CONDITION HIGH
```

Description
- This command is valid when `:SEARch<x>:TYPE EQUalify|SPATtern|STATe`.
- This command is valid when `:SEARch<x>:TYPE WIDTh` and `:SEARch<x>:WIDTh:TYPE PQUalify|PSTAte`.

:SEARch<x>:TRACe<x>:HYSTeresis

Function Sets the hysteresis of the trace or queries the current setting.

Syntax
```
:SEARch<x>:TRACe<x>:HYSTeresis {<NRf>}
:SEARch<x>:TRACe<x>:HYSTeresis?
```
<x> of SEARch<x> = 1 or 2
<x> of TRACe<x> = 1 to 8
<NRf> = 0 to 4 (div, 0.1 div steps)

Example
```
:SEARCH1:TRACE1:HYSTERESIS 1
:SEARCH1:TRACE1:HYSTERESIS? ->
:SEARCH1:TRACE1:HYSTERESIS 1.000E+00
```

:SEARch<x>:TRACe<x>:LEVel

Function Sets the threshold level of the trace or queries the current setting.

Syntax
```
:SEARch<x>:TRACe<x>:LEVel {<NRf>|
<Voltage>|<Current>}
:SEARch<x>:TRACe<x>:LEVel?
```
<x> of SEARch<x> = 1 or 2
<x> of TRACe<x> = 1 to 8
<NRf>, <Voltage>, and <Current> = See the DL9000 User's Manual.

Example
```
:SEARCH1:TRACE1:LEVEL 0
:SEARCH1:TRACE1:LEVEL? ->
:SEARCH1:TRACE1:LEVEL 0.000E+00
```

:SEARch<x>:TYPE

Function Sets the search type or queries the current setting.

Syntax
```
:SEARch<x>:TYPE {EDGE|EQUalify|
Example :SEARCH1:TYPE EDGE
:SEARCH1:TYPE? -> :SEARCH1:TYPE EDGE
```

:SEARch<x>:WIDTh?

Function Queries all settings related to the pulse width search.

Syntax
```
:SEARch<x>:WIDTh?
```
<x> = 1 or 2

Example
```
:SEARCH1:WIDTH? -> :SEARCH1:WIDTH:
MODE OUT;TIME1 1.000E-09;
TIME2 2.000E-09;TYPE PULSE
```

:SEARch<x>:WIDTh:MODE

Function Sets the pulse width determination mode or queries the current setting.

Syntax
```
:SEARch<x>:WIDTh:MODE
{BETWeen|IN|NOTBetween|OUT|TIMeout}
:SEARch<x>:WIDTh:MODE?
```
<x> = 1 or 2

Example
```
:SEARCH1:WIDTH:MODE TIMEOUT
:SEARCH1:WIDTH:MODE? ->
:SEARCH1:WIDTH:MODE TIMEOUT
```

:SEARch<x>:WIDTh:TIME<x>

Function Sets the pulse width of the pulse width search or queries the current setting.

Syntax
```
:SEARch<x>:WIDTh:TIME<x> {<Time>}
:SEARch<x>:WIDTh:TIME<x>?
```
<x> of SEARch<x> = 1 or 2
<x> of TIME<x> = 1 or 2
<Time> = 1 ns to 10 s (500 ps steps)

Example
```
:SEARCH1:WIDTH:TIME1 1S
:SEARCH1:WIDTH:TIME1? ->
:SEARCH1:WIDTH:TIME1 1.000E+00
```

Description TIME2 is valid when `:SEARch<x>:WIDTh:MODE BETWeen|NOTBetween`.

:SEARch<x>:WIDTh:TYPE

Function Sets the pulse width search type or queries the current setting.

Syntax
```
:SEARch<x>:WIDTh:TYPE {PQUalify|PSTAte|
PULSe}
:SEARch<x>:WIDTh:TYPE?
```
<x> = 1 or 2

Example
```
:SEARCH1:WIDTH:TYPE PQUALIFY
:SEARCH1:WIDTH:TYPE? ->
:SEARCH1:WIDTH:TYPE PQUALIFY
```

5.21 SNAP Group

:SNAP

Function Executes the snapshot.

Syntax `:SNAP`

Example `:SNAP`

5.22 SSTart Group

:SSTart?

Function Starts the waveform acquisition with the trigger mode set to single. If the waveform acquisition stops within the specified time period, 0 is returned at that point. If not, 1 is returned.

Syntax `:SSTart? {<NRf>}`

<NRf> = 1 to 360000 (10 ms resolution: wait period, START and wait)
0 (START only. No wait.)
–360000 to –1 (10 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.

5.23 STARt Group

:STARt

Function Starts the waveform acquisition.

Syntax `:STARt`

Example `:START`

Description Use STOP to stop the waveform acquisition.

5.24 STATus Group

The commands in the STATus group are used to make settings and inquiries related to the communication status function. There are no front panel keys that correspond to the commands in this group. For a description of the status report, see chapter 6.

: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 1;QMESSAGE 1

:STATus:CONDition?

Function Queries the contents of the condition register.
Syntax :STATus:CONDition?
Example :STATUS:CONDITION -> 16
Description For details on the condition register, see chapter 6, "Status Report."

:STATus:EESE

Function Sets the extended event enable register or queries the current setting.
Syntax :STATus:EESE <Register>
:STATus:EESE?
<Register> = 0 to 65535
Example :STATUS:EESE 257
:STATUS:EESE? -> :STATUS:EESE 257
Description For details on the extended event enable register, see chapter 6, "Status Report."

:STATus:EESR?

Function Queries the content of the extended event register and clears the register.
Syntax :STATus:EESR?
Example :STATUS:EESR? -> 1
Description For details on the extended event register, see chapter 6, "Status Report."

:STATus:ERRor?

Function Queries the error code and message information (top of the error queue).
Syntax :STATus:ERRor?
Example :STATUS:ERROR? ->
113,"Undefined header"

: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 For details on the transition filter, see chapter 6, "Status Report."

:STATus:QENable

Function Sets whether to store messages other than errors to the error queue or queries the current setting.
Syntax :STATus:QENable {<Boolean>}
:STATus:QENable?
Example :STATUS:QENABLE ON
:STATUS:QENABLE? -> :STATUS:QENABLE 1

:STATus:QMESsage

Function Sets whether or not to attach message information to the response to the "STATus:ERRor?" query or queries the current setting.
Syntax :STATus:QMESsage {<Boolean>}
:STATus:QMESsage?
Example :STATUS:QMESSAGE OFF
:STATUS:QMESSAGE? -> :STATUS:QMESSAGE 0

:STATus:SPOLl? (Serial Poll)

Function Executes serial polling.
Syntax :STATus:SPOLl?
Example :STATUS:SPOLL? -> STATUS:SPOLL 0
Description This command is dedicated to the Ethernet interface (option).

5.25 STOP Group

:STOP

Function Stops the waveform acquisition.

Syntax :STOP

Example :STOP

Description Use STARt to start the waveform acquisition.

5.26 SYSTem Group

:SYSTem?

Function Queries all settings related to the system.
Syntax :SYSTem?
Example :SYSTEM? -> :SYSTEM:CLICK 1;CLOCK:DTIME "2005/05/06","11:37:32","09:00";MODE 1;:SYSTEM:LANGUAGE JAPANESE;MLANGUAGE ENGLISH

:SYSTem:BATTery?

Function Queries the condition of the internal lithium battery.
Syntax :SYSTem:BATTery?
Example :SYSTEM:BATTERY? -> :SYSTEM:BATTERY 1
Description If the lithium battery is normal, 1 is returned. If the battery is dead, 0 is returned.

: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, time, and time difference with respect to GMT.
Syntax :SYSTem:CLOCk?
Example :SYSTEM:CLOCK? -> :SYSTEM:CLOCK:DTIME "2005/05/06","11:37:32","09:00";MODE 1

:SYSTem:CLOCk:DTIMe

Function Sets the date, time, and time difference with respect to GMT or queries the current setting.
Syntax :SYSTem:CLOCk:DTIMe {<String>,<String>,<String>}
:SYSTem:CLOCk:DTIMe?
The left <String> = YYYY/MM/DD. See the DL9000 User's Manual.
The center <String> = HH:MM:SS. See the DL9000 User's Manual.
The right <String> = HH:MM. See the DL9000 User's Manual.
Example :SYSTEM:CLOCK:DTIME "2005/05/06","11:37:32","09:00"
:SYSTEM:CLOCK:DTIME? ->
:SYSTEM:CLOCK:DTIME "2005/05/06","11:37:32","09:00"

:SYSTem:CLOCk:MODE

Function Turns ON/OFF the date, time, and time difference with respect to GMT 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:FORMat:IMEMory[:EXECute]

Function Formats the internal memory.
Syntax :SYSTem:FORMat:IMEMory[:EXECute]
Example :SYSTEM:FORMAT:IMEMORY:EXECUTE

:SYSTem:FORMat:IHDD[:EXECute]

Function Formats the internal hard disk.
Syntax :SYSTem:FORMat:IHDD[:EXECute]
Example :SYSTEM:FORMAT:IHDD:EXECUTE

:SYSTem:FORMat:SDELete[:EXECute] (Sure Delete)

Function Clears and formats the internal memory.
Syntax :SYSTem:FORMat:SDELete[:EXECute]
Example :SYSTEM:FORMAT:SDELETE:EXECUTE

:SYSTem:LANGuage

Function Sets the message language or queries the current setting.
Syntax :SYSTem:LANGuage {CHINese|ENGLish|JAPANese}
:SYSTem:LANGuage?
Example :SYSTEM:LANGUAGE JAPANESE
:SYSTEM:LANGUAGE? ->
:SYSTEM:LANGUAGE JAPANESE

:SYSTem:MLANguage

Function Sets the menu language or queries the current setting.
Syntax :SYSTem:MLANguage {CHINese|ENGLish}
:SYSTem:MLANguage?
Example :SYSTEM:MLANGUAGE ENGLISH
:SYSTEM:MLANGUAGE? ->
:SYSTEM:MLANGUAGE ENGLISH

:SYSTem:OVERview

Function Displays system information.
Syntax :SYSTem:OVERview
Example :SYSTEM:OVERVIEW

5.27 TELecomtest Group

:TELecomtest?

Function Queries all settings related to the telecom test.

Syntax :TELecomtest?

Example
```
:TELECOMTEST? -> :TELECOMTEST:
CATEGORY MASK;DISPLAY 0;EYEPATTERN:
DBERATE:STATE 0;:TELECOMTEST:
EYEPATTERN:EHEIGHT:STATE 0;:
TELECOMTEST:EYEPATTERN:EWIDTH:STATE 0;:
TELECOMTEST:EYEPATTERN:FALL:STATE 0;:
TELECOMTEST:EYEPATTERN:JITTER:STATE 0;:
TELECOMTEST:EYEPATTERN:PCROSSING:
STATE 0;:TELECOMTEST:EYEPATTERN:
PDUTYCYCLE:STATE 0;:TELECOMTEST:
EYEPATTERN:QFACTOR:STATE 0;:
TELECOMTEST:EYEPATTERN:RISE:STATE 0;:
TELECOMTEST:EYEPATTERN:SDBASE:STATE 0;:
TELECOMTEST:EYEPATTERN:SDTOP:STATE 0;:
TELECOMTEST:EYEPATTERN:T1CROSSING:
STATE 0;:TELECOMTEST:EYEPATTERN:
T2CROSSING:STATE 0;:TELECOMTEST:
EYEPATTERN:TLEVELS:MODE PERCENT;
PERCENT 90,10;
UNIT 1.000E+00,0.000E+00;:TELECOMTEST:
EYEPATTERN:VBASE:STATE 0.....
```

:TELecomtest:CATegory

Function Sets the telecom test type or queries the current setting.

Syntax
```
:TELecomtest:CATegory {EYEPattern|MASK}
:TELecomtest:CATegory?
```

Example
```
:TELECOMTEST:CATEGORY EYEPATTERN
:TELECOMTEST:CATEGORY? ->
:TELECOMTEST:CATEGORY EYEPATTERN
```

:TELecomtest:DISPlay

Function Turns ON/OFF the telecom test display or queries the current setting.

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

Example
```
:TELECOMTEST:DISPLAY ON
:TELECOMTEST:DISPLAY? ->
:TELECOMTEST:DISPLAY 1
```

:TELecomtest:EYEPattern?

Function Queries all settings related to the eye pattern.

Syntax :TELecomtest:EYEPattern?

Example
```
:TELECOMTEST:EYEPATTERN? ->
:TELECOMTEST:EYEPATTERN:DBERATE:
STATE 1;:TELECOMTEST:EYEPATTERN:
EHEIGHT:STATE 1;:TELECOMTEST:
EYEPATTERN:EWIDTH:STATE 1;:
TELECOMTEST:EYEPATTERN:FALL:STATE 1;:
TELECOMTEST:EYEPATTERN:JITTER:STATE 1;:
EYEPATTERN:QFACTOR:STATE 1;:
TELECOMTEST:EYEPATTERN:RISE:STATE 1;:
TELECOMTEST:EYEPATTERN:SDBASE:STATE 1;:
TELECOMTEST:EYEPATTERN:SDTOP:STATE 1;:
TELECOMTEST:EYEPATTERN:T1CROSSING:
STATE 1;:TELECOMTEST:EYEPATTERN:
T2CROSSING:STATE 1;:TELECOMTEST:
EYEPATTERN:TLEVELS:MODE PERCENT;
PERCENT 90,10;
UNIT 1.000E+00,0.000E+00;:TELECOMTEST:
EYEPATTERN:VBASE:STATE 1;:TELECOMTEST:
EYEPATTERN:VCROSSING:STATE 1;:
TELECOMTEST:EYEPATTERN:VDARK 1;VTOP:
STATE 1
```

:TELecomtest:EYEPattern:ALL

Function Turns ON/OFF all eye pattern parameters.

Syntax :TELecomtest:EYEPattern:ALL {<Boolean>}

Example :TELECOMTEST:EYEPATTERN:ALL ON

:TELecomtest:EYEPattern:<Parameter>?

Function Queries all settings related to the waveform parameter of the eye pattern.

Syntax
```
:TELecomtest:EYEPattern:<Parameter>?
<Parameter> = {DBERate|EHEight|EWIDth|
FALL|JITTer|PCROssing|PDUTycycle|
QFACtor|RISE|SDBase|SDTop|T1CRossing|
T2CRossing|VBASe|VCRossing|VTOP}
```

Example (The following is an example for DBERate.)
```
:TELECOMTEST:EYEPATTERN:DBERATE? ->
:TELECOMTEST:EYEPATTERN:DBERATE:STATE 1
```

Description
- For the relationship between communication commands and parameters used by the DL9000, see appendix 4.
- For parameter details, see the DL9000 User's Manual.

:TELecomtest:EYEPattern:<Parameter>:STATe

Function Turns ON/OFF the waveform parameter of the eye pattern or queries the current setting.

Syntax
```
:TELecomtest:EYEPattern:<Parameter>:
STATe {<Boolean>}
:TELecomtest:EYEPattern:<Parameter>:
STATe?
```
<Parameter> = {DBERate|EHEight|EWIDth|FALL|JITTer|PCROssing|PDUTycycle|QFACtor|RISESDBase|SDTop|T1CRossing|T2CRossing|VBASe|VCRossing|VTOP}

Example (The following is an example for DBERate.)
```
:TELECOMTEST:EYEPATTERN:DBERATE:
STATE ON
:TELECOMTEST:EYEPATTERN:DBERATE:STATE?
-> :TELECOMTEST:EYEPATTERN:DBERATE:
STATE 1
```

:TELecomtest:EYEPattern:<Parameter>:VALue?

Function Queries the waveform parameter value of the eye pattern.

Syntax
```
:TELecomtest:EYEPattern:<Parameter>:
VALue?
```
<Parameter> = {DBERate|EHEight|EWIDth|FALL|JITTer|PCROssing|PDUTycycle|QFACtor|RISESDBase|SDTop|T1CRossing|T2CRossing|VBASe|VCRossing|VTOP}

Example (The following is an example for DBERate.)
```
:TELECOMTEST:EYEPATTERN:DBERATE:VALUE?
-> :TELECOMTEST:EYEPATTERN:DBERATE:
VALUE 1.000E+00
```

:TELecomtest:EYEPattern:TLEVels?

Function Queries all settings related to the threshold level of the eye pattern.

Syntax
```
:TELecomtest:EYEPattern:TLEVels?
```

Example
```
:TELECOMTEST:EYEPATTERN:TLEVELS? ->
:TELECOMTEST:EYEPATTERN:TLEVELS:
MODE PERCENT;PERCENT 90,10;
UNIT 1.000E+00,0.000E+00
```

:TELecomtest:EYEPattern:TLEVels:MODE

Function Sets the unit of the threshold level of the eye pattern or queries the current setting.

Syntax
```
:TELecomtest:EYEPattern:TLEVels:
MODE {PERCent|UNIT}
:TELecomtest:EYEPattern:TLEVels:MODE?
```

Example
```
:TELECOMTEST:EYEPATTERN:TLEVELS:
MODE PERCENT
:TELECOMTEST:EYEPATTERN:TLEVELS:MODE?
-> :TELECOMTEST:EYEPATTERN:TLEVELS:
MODE PERCENT
```

:TELecomtest:EYEPattern:TLEVels:PERCent

Function Sets the threshold level of the eye pattern as a percentage or queries the current setting.

Syntax
```
:TELecomtest:EYEPattern:TLEVels:
PERCent {<NRf>,<NRf>}
:TELecomtest:EYEPattern:TLEVels:
PERCent?
```
<NRf> = 0 to 100 (%)

Example
```
:TELECOMTEST:EYEPATTERN:TLEVELS:
PERCENT 90,10
:TELECOMTEST:EYEPATTERN:TLEVELS:
PERCENT? -> :TELECOMTEST:EYEPATTERN:
TLEVELS:PERCENT 90,10
```

:TELecomtest:EYEPattern:TLEVels:UNIT

Function Sets the threshold level of the eye pattern in UNIT or queries the current setting.

Syntax
```
:TELecomtest:EYEPattern:TLEVels:
UNIT {<NRf>,<NRf>|<Voltage>,<Voltage>|
<Current>,<Current>}
:TELecomtest:EYEPattern:TLEVels:UNIT?
```
<NRf>, <Voltage>, and <Current> = See the DL9000 User's Manual.

Example
```
:TELECOMTEST:EYEPATTERN:TLEVELS:
UNIT 1,0
:TELECOMTEST:EYEPATTERN:TLEVELS:UNIT?
-> :TELECOMTEST:EYEPATTERN:TLEVELS:
UNIT 1.000E+00,0.000E+00
```

:TELecomtest:EYEPattern:VDARk

Function Sets the dark level (zero light level) or queries the current setting.

Syntax
```
:TELecomtest:EYEPattern:VDARk {<NRf>|
<Voltage>|<Current>}
:TELecomtest:EYEPattern:VDARk?
```
<NRf>, <Voltage>, and <Current> = See the DL9000 User's Manual.

Example
```
:TELECOMTEST:EYEPATTERN:VDARK 1.000E+00
:TELECOMTEST:EYEPATTERN:VDARK? ->
:TELECOMTEST:EYEPATTERN:VDARK 1.000E+00
```

:TELecomtest:MASK?

Function Queries all settings related to the mask test.

Syntax
```
:TELecomtest:MASK?
```

Example
```
:TELECOMTEST:MASK? -> :TELECOMTEST:
MASK:ELEMENT1:PSPCOUNT:STATE 1;:
TELECOMTEST:MASK:ELEMENT1:PWCOUNT:
STATE 1;:TELECOMTEST:MASK:ELEMENT1:
SPCOUNT:STATE 1;:TELECOMTEST:MASK:
ELEMENT1:WCOUNT:STATE 1;:TELECOMTEST:
MASK:ELEMENT2:PSPCOUNT:STATE 1;:
TELECOMTEST:MASK:ELEMENT2:PWCOUNT:
STATE 1;:TELECOMTEST:MASK:ELEMENT2:
SPCOUNT:STATE 1;:TELECOMTEST:MASK:
ELEMENT2:WCOUNT:STATE 1;:TELECOMTEST:
MASK:ELEMENT3:PSPCOUNT:STATE 1;:
TELECOMTEST:MASK:ELEMENT3:PWCOUNT:
STATE 1;:TELECOMTEST:MASK:ELEMENT3:
SPCOUNT:STATE 1;:TELECOMTEST:MASK:
ELEMENT3:WCOUNT:STATE 1;:TELECOMTEST:
MASK:ELEMENT4:PSPCOUNT:STATE 1;:
TELECOMTEST:MASK:ELEMENT4:PWCOUNT:
STATE 1;:TELECOMTEST:MASK:ELEMENT4:
SPCOUNT:STATE 1;:TELECOMTEST:MASK:
ELEMENT4:WCOUNT:STATE 1
```

:TELecomtest:MASK:ELEMent<x>?

Function Queries all settings related to the element used in the mask test.

Syntax
```
:TELecomtest:MASK:ELEMent<x>?
```
<x> = 1 to 4

Example
```
:TELECOMTEST:MASK:ELEMENT1? ->
:TELECOMTEST:MASK:ELEMENT1:PSPCOUNT:
STATE 1;:TELECOMTEST:MASK:ELEMENT1:
PWCOUNT:STATE 1;:TELECOMTEST:MASK:
ELEMENT1:SPCOUNT:STATE 1;:TELECOMTEST:
MASK:ELEMENT1:WCOUNT:STATE 1
```

:TELecomtest:MASK:ELEMent<x>:ALL

Function Turns ON/OFF all items of the element.

Syntax
```
:TELecomtest:MASK:ELEMent<x>:ALL
{<Boolean>}
```
<x> = 1 to 4

Example
```
:TELECOMTEST:MASK:ELEMENT1:ALL ON
```

:TELecomtest:MASK:ELEMent<x>: PSPCount? (Sample Point Count %)

Function Queries the settings related to the error rate for the number of sampled data points of the element.

Syntax
```
:TELecomtest:MASK:ELEMent<x>:PSPCount?
```
<x> = 1 to 4

Example
```
:TELECOMTEST:MASK:ELEMENT1:PSPCOUNT? ->
:TELECOMTEST:MASK:ELEMENT1:PSPCOUNT:
STATE 1
```

:TELecomtest:MASK:ELEMent<x>: PSPCount:STATe

Function Turns ON/OFF the measurement of the error rate for the number of sampled data points of the element or queries the current setting.

Syntax
```
:TELecomtest:MASK:ELEMent<x>:PSPCount:
STATe {<Boolean>}
:TELecomtest:MASK:ELEMent<x>:PSPCount:
STATe?
```
<x> = 1 to 4

Example
```
:TELECOMTEST:MASK:ELEMENT1:PSPCOUNT:
STATE ON
:TELECOMTEST:MASK:ELEMENT1:PSPCOUNT:
STATE? -> :TELECOMTEST:MASK:ELEMENT1:
PSPCOUNT:STATE 1
```

:TELecomtest:MASK:ELEMent<x>: PSPCount:VALue?

Function Queries the error rate for the number of sampled data points of the element.

Syntax
```
:TELecomtest:MASK:ELEMent<x>:PSPCount:
VALue?
```
<x> = 1 to 4

Example
```
:TELECOMTEST:MASK:ELEMENT1:PSPCOUNT:
VALUE? -> :TELECOMTEST:MASK:ELEMENT1:
PSPCOUNT:VALUE 1.000E+00
```

:TELecomtest:MASK:ELEMent<x>: PWCount? (Wave Count %)

Function Queries the settings related to the error rate for the acquisition count of the element.

Syntax
```
:TELecomtest:MASK:ELEMent<x>:PWCount?
```
<x> = 1 to 4

Example
```
:TELECOMTEST:MASK:ELEMENT1:PWCOUNT? ->
:TELECOMTEST:MASK:ELEMENT1:PWCOUNT:
STATE 1
```

:TELecomtest:MASK:ELEMent<x>:PWCount:STATe

Function Turns ON/OFF the measurement of the error rate for the acquisition count of the element or queries the current setting.

Syntax
```
:TELecomtest:MASK:ELEMent<x>:PWCount:STATe {<Boolean>}
:TELecomtest:MASK:ELEMent<x>:PWCount:STATe?
```
<x> = 1 to 4

Example
```
:TELECOMTEST:MASK:ELEMENT1:PWCOUNT:STATE ON
:TELECOMTEST:MASK:ELEMENT1:PWCOUNT:STATE? -> :TELECOMTEST:MASK:ELEMENT1:PWCOUNT:STATE 1
```

:TELecomtest:MASK:ELEMent<x>:PWCount:VALue?

Function Queries the error rate for the acquisition count of the element.

Syntax
```
:TELecomtest:MASK:ELEMent<x>:PWCount:VALue?
```
<x> = 1 to 4

Example
```
:TELECOMTEST:MASK:ELEMENT1:PWCOUNT:VALUE? -> :TELECOMTEST:MASK:ELEMENT1:PWCOUNT:VALUE 1.000E+00
```

:TELecomtest:MASK:ELEMent<x>:SPCount? (Sample Point Count)

Function Queries the settings related to the number of sampled data points for the element that results in error.

Syntax
```
:TELecomtest:MASK:ELEMent<x>:SPCount?
```
<x> = 1 to 4

Example
```
:TELECOMTEST:MASK:ELEMENT1:SPCOUNT? -> :TELECOMTEST:MASK:ELEMENT1:SPCOUNT:STATE 1
```

:TELecomtest:MASK:ELEMent<x>:SPCount:STATe

Function Turns ON/OFF the measurement of the number of sampled data points for the element that results in error or queries the current setting.

Syntax
```
:TELecomtest:MASK:ELEMent<x>:SPCount:STATe {<Boolean>}
:TELecomtest:MASK:ELEMent<x>:SPCount:STATe?
```
<x> = 1 to 4

Example
```
:TELECOMTEST:MASK:ELEMENT1:SPCOUNT:STATE ON
:TELECOMTEST:MASK:ELEMENT1:SPCOUNT:STATE? -> :TELECOMTEST:MASK:ELEMENT1:SPCOUNT:STATE 1
```

:TELecomtest:MASK:ELEMent<x>:SPCount:VALue?

Function Queries the number of sampled data points for the element that resulted in error.

Syntax
```
:TELecomtest:MASK:ELEMent<x>:SPCount:VALue?
```
<x> = 1 to 4

Example
```
:TELECOMTEST:MASK:ELEMENT1:SPCOUNT:VALUE? -> :TELECOMTEST:MASK:ELEMENT1:SPCOUNT:VALUE 1
```

:TELecomtest:MASK:ELEMent<x>:WCOunt? (Wave Count)

Function Queries the settings related to the acquisition count for the element that results in error.

Syntax
```
:TELecomtest:MASK:ELEMent<x>:WCOunt?
```
<x> = 1 to 4

Example
```
:TELECOMTEST:MASK:ELEMENT1:WCOUNT? -> :TELECOMTEST:MASK:ELEMENT1:WCOUNT:STATE 1
```

:TELecomtest:MASK:ELEMent<x>:WCOunt:STATe

Function Turns ON/OFF the measurement of the acquisition count for the element that results in error or queries the current setting.

Syntax
```
:TELecomtest:MASK:ELEMent<x>:WCOunt:STATe {<Boolean>}
:TELecomtest:MASK:ELEMent<x>:WCOunt:STATe?
```
<x> = 1 to 4

Example
```
:TELECOMTEST:MASK:ELEMENT1:WCOUNT:STATE ON
:TELECOMTEST:MASK:ELEMENT1:WCOUNT:STATE? -> :TELECOMTEST:MASK:ELEMENT1:WCOUNT:STATE 1
```

:TELecomtest:MASK:ELEMent<x>:WCOunt:VALue?

Function Queries the acquisition count for the element that resulted in error.

Syntax
```
:TELecomtest:MASK:ELEMent<x>:WCOunt:VALue?
```
<x> = 1 to 4

Example
```
:TELECOMTEST:MASK:ELEMENT1:WCOUNT:VALUE? -> :TELECOMTEST:MASK:ELEMENT1:WCOUNT:VALUE 1
```

:TELecomtest:MMODe

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

```
Syntax  :TELecomtest:MMODe {<Boolean>}
        :TELecomtest:MMODe?
Example :TELECOMTEST:MMODE ON
        :TELECOMTEST:MMODE? ->
        :TELECOMTEST:MMODE 1
```

:TELecomtest:TRACe

Function Sets the source trace of the telecom test or queries the current setting.

```
Syntax  :TELecomtest:TRACe {<NRf>}
        :TELecomtest:TRACe?
        <NRf> = 1 to 8
Example :TELECOMTEST:TRACE 1
        :TELECOMTEST:TRACE? ->
        :TELECOMTEST:TRACE 1
```

:TELecomtest:TRANge (Time Range)

Function Sets the measurement range of the telecom test or queries the current setting.

```
Syntax  :TELecomtest:TRANge {<NRf>,<NRf>}
        :TELecomtest:TRANge?
        <NRf> = –5 to 5 (div)
Example :TELECOMTEST:TRANGE -5,0
        :TELECOMTEST:TRANGE? -> :TELECOMTEST:
        TRANGE 0.000E+00,-5.000E+00
```

:TELecomtest:WINDow

Function Sets the measurement source window of the telecom test or queries the current setting.

```
Syntax  :TELecomtest:WINDow {MAIN|Z1|Z2}
        :TELecomtest:WINDow?
Example :TELECOMTEST:WINDOW MAIN
        :TELECOMTEST:WINDOW? ->
        :TELECOMTEST:WINDOW MAIN
```

5.28 TIMebase Group

:TIMebase?

Function Queries all settings related to the time base.

```
Syntax   :TIMebase?
Example  :TIMEBASE? -> :TIMEBASE:TDIV 1.000E-06
```

:TIMebase:SRATe? (Sample RATE)

Function Queries the sample rate or queries the current setting.

```
Syntax   :TIMebase:SRATe?
Example  :TIMEBASE:SRATE? ->
         :TIMEBASE:SRATE 125.0E+06
```

:TIMebase:TDIV

Function Sets the T/div value or queries the current setting.

```
Syntax   :TIMebase:TDIV {<Time>}
         :TIMebase:TDIV?
```

<Time> = 500 ps to 50 s

```
Example  :TIMEBASE:TDIV 1NS
         :TIMEBASE:TDIV? ->
         :TIMEBASE:TDIV 1.000E-06
```

5.29 TRIGger Group

:TRIGger?

Function Queries all settings related to the trigger.

Syntax :TRIGger?

Example :TRIGGER? -> :TRIGGER:ACTION:
ACQCOUNT 1;BUZZER 0;HCOPY 1;
MODE ACONDITION;SAVE 1;:TRIGGER:
TYPE EICYCLE;CLOCK:SOURCE 1;
POLARITY RISE;:TRIGGER:DELAY:EDGECOUNT:
COUNT 1;:TRIGGER:DELAY:MODE 1;
POLARITY FALL;SOURCE 4;TIME 1.000E+00;
TYPE EDGECOUNT;:TRIGGER:EINTERVAL:
EVENT1:TYPE EDGE;CLOCK:SOURCE 1;
POLARITY FALL;:TRIGGER:EINTERVAL:
EVENT1:ESTATE:SOURCE 1;POLARITY FALL;:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT10ADDRESS:PATTERN "10111011111";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN "11011110";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "11001101";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
TYPE BIT10ADDRESS;:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:CLOCK:SOURCE 1;:TRIGGER:
EINTERVAL:EVENT1:I2CBUS:DATA:BYTE 1;
CONDITION TRUE;DPOSITION 1;MODE 1;
PATTERN1 "10101011";
PATTERN2 "10101010";
PATTERN3 "10101111";
PATTERN4 "10101011";PMODE DONTCARE;
SOURCE 1;:TRIGGER:EINTERVAL:EVENT1:
I2CBUS:GCALL:BIT7MADDRESS:
PATTERN "1010101";:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:GCALL:
SBYTE BIT7MADDRESS;:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:MODE ADATA;NAIGNORE:
HSMODE 1;RACCESS 1;SBYTE 1;:TRIGGER:
EINTERVAL:EVENT1:I2CBUS:SBHSMODE:
TYPE HSMODE;:TRIGGER:EINTERVAL:EVENT1:
SPATTERN:BITRATE 1.000E+00;CLOCK:
MODE 1;POLARITY FALL;SOURCE 1;:
TRIGGER:EINTERVAL:EVENT1:SPATTERN:CS 1;
DATA:ACTIVE HIGH;SOURCE 1;:TRIGGER:
EINTERVAL:EVENT1:SPATTERN:LATCH:
SOURCE 1;POLARITY FALL;:TRIGGER:
EINTERVAL:EVENT1:SPATTERN:
PATTERN "1100110111101111".....

:TRIGger:ACTion?

Function Queries all settings related to action-on-trigger.

Syntax :TRIGger:ACTion?

Example :TRIGGER:ACTION? ->
:TRIGGER:ACTION:ACQCOUNT 1;BUZZER 0;
HCOPY 1;MODE ACONDITION;SAVE 1

:TRIGger:ACTion:ACQCount

Function Sets the action count of action-on-trigger or queries the current setting.

Syntax :TRIGger:ACTion:ACQCount {<NRf>|INFinite}
:TRIGger:ACTion:ACQCount?
<NRf> = 1 to 1000000

Example :TRIGGER:ACTION:ACQCOUNT 10
:TRIGGER:ACTION:ACQCOUNT? ->
:TRIGGER:ACTION:ACQCOUNT 10

:TRIGger:ACTion:BUZZer

Function Sets whether to sound a buzzer when an 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 or not to output screen image data (ON/OFF) 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:MODE

Function Sets the action-on-trigger mode or queries the current setting.

Syntax :TRIGger:ACTion:MODE {ACONdition|OFF}
:TRIGger:ACTion:MODE?

Example :TRIGGER:ACTION:MODE ACONDITION
:TRIGGER:ACTION:MODE? ->
:TRIGGER:ACTION:MODE ACONDITION

:TRIGger:ACTion:SAVE

Function Sets whether 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:STARt

Function Starts the action-on-trigger.

Syntax
```
:TRIGger:ACTion:STARt
```

Example
```
:TRIGGER:ACTION:START
```

:TRIGger:ACTion:STOP

Function Stops the action-on-trigger.

Syntax
```
:TRIGger:ACTion:STOP
```

Example
```
:TRIGGER:ACTION:STOP
```

:TRIGger:CLOCk?

Function Queries all settings related to the clock channel.

Syntax
```
:TRIGger:CLOCk?
```

Example
```
:TRIGGER:CLOCK? ->
:TRIGGER:CLOCK:SOURCE 1;POLARITY RISE
```

:TRIGger:CLOCk:POLarity

Function Sets the polarity of the clock channel or queries the current setting.

Syntax
```
:TRIGger:CLOCk:POLarity {ENTer|EXIT|FALL|RISE}
:TRIGger:CLOCk:POLarity?
```

Example
```
:TRIGGER:CLOCK:POLARITY RISE
:TRIGGER:CLOCK:POLARITY? ->
:TRIGGER:CLOCK:POLARITY RISE
```

Description
- This command is invalid when `:TRIGger:CLOCk:SOURce NONE`.
- `{ENTer|EXIT}` is valid when `:TRIGger:SOURce:CHANnel<x>:WINDow ON`. For all other cases, `{FALL|RISE}` is valid.
- This command is valid when `:TRIGger:TYPE PSTAte|STATe`.

:TRIGger:CLOCk:SOURce

Function Sets the source waveform of the clock channel or queries the current setting.

Syntax
```
:TRIGger:CLOCk:SOURce {<NRf>|NONE}
:TRIGger:CLOCk:SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:CLOCK:SOURCE NONE
:TRIGGER:CLOCK:SOURCE? ->
:TRIGGER:CLOCK:SOURCE NONE
```

Description This command is valid when `:TRIGger:TYPE PSTAte|STATe`.

:TRIGger:DELay?

Function Queries all settings related to the trigger delay.

Syntax
```
:TRIGger:DELay?
```

Example
```
:TRIGGER:DELAY? ->
:TRIGGER:DELAY:EDGECOUNT:COUNT 1;:
TRIGGER:DELAY:MODE 1;POLARITY FALL;
SOURCE 4;TIME 1.000E+00;TYPE EDGECOUNT
```

:TRIGger:DELay:EDGecount?

Function Queries all settings related to edge count of the trigger delay.

Syntax
```
:TRIGger:DELay:EDGecount?
```

Example
```
:TRIGGER:DELAY:EDGECOUNT? ->
:TRIGGER:DELAY:EDGECOUNT:COUNT 1
```

:TRIGger:DELay:EDGecount:COUNt

Function Sets the edge count value of the trigger delay or queries the current setting.

Syntax
```
:TRIGger:DELay:EDGecount:COUNt {<NRf>}
:TRIGger:DELay:EDGecount:COUNt?
```
<NRf> = 1 to 1000000000

Example
```
:TRIGGER:DELAY:EDGECOUNT:COUNT 1
:TRIGGER:DELAY:EDGECOUNT:COUNT? ->
:TRIGGER:DELAY:EDGECOUNT:COUNT 1
```

:TRIGger:DELay:MODE

Function Turns ON/OFF the trigger delay or queries the current setting.

Syntax
```
:TRIGger:DELay:MODE {<Boolean>}
:TRIGger:DELay:MODE?
```

Example
```
:TRIGGER:DELAY:MODE ON
:TRIGGER:DELAY:MODE? ->
:TRIGGER:DELAY:MODE 1
```

:TRIGger:DELay:POLarity

Function Sets the edge polarity the trigger delay or queries the current setting.

Syntax
```
:TRIGger:DELay:POLarity {FALL|RISE}
:TRIGger:DELay:POLarity?
```

Example
```
:TRIGGER:DELAY:POLARITY RISE
:TRIGGER:DELAY:POLARITY? ->
:TRIGGER:DELAY:POLARITY RISE
```

Description This command is valid when `:TRIGger:DELay:TYPE EDGecount|FEADelay`.

:TRIGger:DELay:SOURce

Function Sets the edge source the trigger delay or queries the current setting.

Syntax
```
:TRIGger:DELay:SOURce {<NRf>|EXTernal}
:TRIGger:DELay:SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:DELAY:SOURCE 1
:TRIGGER:DELAY:SOURCE? ->
:TRIGGER:DELAY:SOURCE 1
```

Description This command is valid when
```
:TRIGger:DELay:TYPE EDGecount|FEADelay.
```

:TRIGger:DELay:TIME

Function Sets the delay value the trigger delay or queries the current setting.

Syntax
```
:TRIGger:DELay:TIME {<Time>}
:TRIGger:DELay:TIME?
```
<Time> = 0 s to 10 s (5 ps steps)

Example
```
:TRIGGER:DELAY:TIME 1S
:TRIGGER:DELAY:TIME? ->
:TRIGGER:DELAY:TIME 1.000E+00
```

Description This command is valid when
```
:TRIGger:DELay:TYPE BYTime|FEADelay.
```

:TRIGger:DELay:TYPE

Function Sets the trigger delay type or queries the current setting.

Syntax
```
:TRIGger:DELay:TYPE {BYTime|EDGecount|
FEADelay}
:TRIGger:DELay:TYPE?
```

Example
```
:TRIGGER:DELAY:TYPE BYTIME
:TRIGGER:DELAY:TYPE? ->
:TRIGGER:DELAY:TYPE BYTIME
```

:TRIGger:EINTerval?

Function Queries all settings related to the event interval.

Syntax
```
:TRIGger:EINTerval?
```

Example
```
:TRIGGER:EINTERVAL? -> :TRIGGER:
EINTERVAL:EVENT1:TYPE EDGE;CLOCK:
SOURCE 1;POLARITY FALL;:TRIGGER:
EINTERVAL:EVENT1:ESTATE:SOURCE 1;
POLARITY FALL;:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:ADATA:BIT10ADDRESS:
PATTERN "10111011111";:TRIGGER:
EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN "11011110";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "11001101";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
TYPE BIT10ADDRESS;:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:CLOCK:SOURCE 1;:TRIGGER:
EINTERVAL:EVENT1:I2CBUS:DATA:BYTE 1;
CONDITION TRUE;DPOSITION 1;MODE 1;
PATTERN1 "10101011";
PATTERN2 "10101010";
PATTERN3 "10101111";
PATTERN4 "10101011";PMODE DONTCARE;
SOURCE 1;:TRIGGER:EINTERVAL:EVENT1:
I2CBUS:GCALL:BIT7MADDRESS:
PATTERN "1010101";:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:GCALL:
SBYTE BIT7MADDRESS;:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:MODE ADATA;NAIGNORE:
HSMODE 1;RACCESS 1;SBYTE 1;:TRIGGER:
EINTERVAL:EVENT1:I2CBUS:SBHSMODE:
TYPE HSMODE;:TRIGGER:EINTERVAL:EVENT1:
SPATTERN:BITRATE 1.000E+00;CLOCK:
MODE 1;POLARITY FALL;SOURCE 1;:
TRIGGER:EINTERVAL:EVENT1:SPATTERN:CS 1;
DATA:ACTIVE HIGH;SOURCE 1;:TRIGGER:
EINTERVAL:EVENT1:SPATTERN:LATCH:
SOURCE 1;POLARITY FALL;:TRIGGER:
EINTERVAL:EVENT1:SPATTERN:
PATTERN "11001101111101111".....
```

:TRIGger:EINTerval:EVENt<x>?

Function Queries all settings related to the event.

Syntax
```
:TRIGger:EINTerval:EVENt<x>?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1? ->
:TRIGGER:EINTERVAL:EVENT1:TYPE EDGE;
CLOCK:SOURCE 1;POLARITY FALL;:TRIGGER:
EINTERVAL:EVENT1:ESTATE:SOURCE 1;
POLARITY FALL;:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:ADATA:BIT10ADDRESS:
PATTERN "10111011111";:TRIGGER:E
INTERVAL:EVENT1:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN "11011110";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "11001101";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
TYPE BIT10ADDRESS;:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:CLOCK:SOURCE 1;:TRIGGER:
EINTERVAL:EVENT1:I2CBUS:DATA:BYTE 1;
CONDITION TRUE;DPOSITION 1;MODE 1;
PATTERN1 "10101011";
PATTERN2 "10101010";
PATTERN3 "10101111";
PATTERN4 "10101011";PMODE DONTCARE;
SOURCE 1;:TRIGGER:EINTERVAL:EVENT1:
I2CBUS:GCALL:BIT7MADDRESS:
PATTERN "1010101";:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:GCALL:
SBYTE BIT7MADDRESS;:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:MODE ADATA;NAIGNORE:
HSMODE 1;RACCESS 1;SBYTE 1;:TRIGGER:
EINTERVAL:EVENT1:I2CBUS:SBHSMODE:
TYPE HSMODE;:TRIGGER:EINTERVAL:EVENT1:
SPATTERN:BITRATE 1.000E+00;CLOCK:
MODE 1;POLARITY FALL;SOURCE 1;:
TRIGGER:EINTERVAL:EVENT1:SPATTERN:CS 1;
DATA:ACTIVE HIGH;SOURCE 1;:TRIGGER:
EINTERVAL:EVENT1:SPATTERN:LATCH:
SOURCE 1;POLARITY FALL;:TRIGGER:
EINTERVAL:EVENT1:SPATTERN:
PATTERN "1100110111101111".....
```

Description EVENt2 is valid when `:TRIGger:TYPE EIDelay|EISequence`.

:TRIGger:EINTerval:EVENt<x>:CLOCk?

Function Queries all settings related to the clock channel of the event.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CLOCk?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:CLOCK? ->
:TRIGGER:EINTERVAL:EVENT1:CLOCK:
SOURCE 1;POLARITY FALL
```

:TRIGger:EINTerval:EVENt<x>:CLOCk:POLarity

Function Sets the polarity of the clock channel of the event or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CLOCk:
POLarity {ENTer|EXIT|FALL|RISE}
:TRIGger:EINTerval:EVENt<x>:CLOCk:
POLarity?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:CLOCK:
POLARITY FALL
:TRIGGER:EINTERVAL:EVENT1:CLOCK:
POLARITY? -> :TRIGGER:EINTERVAL:EVENT1:
CLOCK:POLARITY FALL
```

Description
- This command in invalid when `:TRIGger:EINTerval:EVENt<x>:CLOCk:SOURce NONE`.
- `{ENTer|EXIT}` is valid when `:TRIGger:SOURce:CHANnel<x>:WINDow ON`. For all other cases, `{FALL|RISE}` is valid.
- This command is valid when `:TRIGger:EINTerval:EVENt<x>:TYPE PSTAte|STATe`.

:TRIGger:EINTerval:EVENt<x>:CLOCk:SOURce

Function Sets the source waveform of the clock channel of the event or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CLOCk:
SOURce {<NRf>|NONE}
:TRIGger:EINTerval:EVENt<x>:CLOCk:
SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:CLOCK:
SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:CLOCK:SOURCE?
-> :TRIGGER:EINTERVAL:EVENT1:CLOCK:
SOURCE 1
```

Description This command is valid when `:TRIGger:EINTerval:EVENt<x>:TYPE PSTAte|STATe`.

:TRIGger:EINTerval:EVENt<x>:ESTate?

Function Queries all settings related to the edge/state trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:ESTate?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:ESTATE? ->
:TRIGGER:EINTERVAL:EVENT1:ESTATE:
SOURCE 1;POLARITY FALL
```

:TRIGger:EINTerval:EVENt<x>:ESTate:POLarity

Function Sets the polarity of the edge/state trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:ESTate:
POLarity {ENTer|EXIT|FALL|RISE}
:TRIGger:EINTerval:EVENt<x>:ESTate:
POLarity?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:ESTATE:
POLARITY ENTER
:TRIGGER:EINTERVAL:EVENT1:ESTATE:
POLARITY? -> :TRIGGER:EINTERVAL:
EVENT1:ESTATE:POLARITY ENTER
```

Description
- This command is valid when :TRIGger:EINTerval:EVENt<x>:TYPE EDGE and :TRIGger:EINTerval:EVENt<x>:ESTate:SOURce LINE.
- {ENTer|EXIT} is valid when :TRIGger:EINTerval:EVENt<x>:TYPE EDGE|EQUalify and :TRIGger:SOURce:CHANnel<x>:WINDow ON.
  {FALL|RISE} is valid when :TRIGger:EINTerval:EVENt<x>:TYPE EDGE|EQUalify and :TRIGger:SOURce:CHANnel<x>:WINDow OFF.
- {ENTer|EXIT} is valid when :TRIGger:EINTerval:EVENt<x>:TYPE STATe.

:TRIGger:EINTerval:EVENt<x>:ESTate:SOURce

Function Sets the trigger source of the edge/state trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:ESTate:
SOURce {<NRf>|EXTernal|LINE}
:TRIGger:EINTerval:EVENt<x>:ESTate:
SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:ESTATE:
SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:ESTATE:
SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:
ESTATE:SOURCE 1
```

Description
- This command is valid when :TRIGger:EINTerval:EVENt<x>:TYPE EDGE|EQUalify.
- {<NRf>|EXTernal|LINE} is valid when :TRIGger:EINTerval:EVENt<x>:TYPE EDGE.
- {<NRf>|EXTernal} is valid when :TRIGger:EINTerval:EVENt<x>:TYPE EQUalify.

:TRIGger:EINTerval:EVENt<x>:I2CBus?

Function Queries all settings related to the I²C bus trigger of the event.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT10ADDRESS:PATTERN "10111011111";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN "11011110";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
TYPE BIT10ADDRESS;:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:CLOCK:SOURCE 1;:TRIGGER:
EINTERVAL:EVENT1:I2CBUS:DATA:BYTE 1;
CONDITION TRUE;DPOSITION 1;MODE 1;
PATTERN1 "10101011";
PATTERN2 "10101010";
PATTERN3 "10101111";
PATTERN4 "10101011";PMODE DONTCARE;
SOURCE 1;:TRIGGER:EINTERVAL:EVENT1:
I2CBUS:GCALL:BIT7MADDRESS:
PATTERN "1010101";:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:GCALL:
SBYTE BIT7MADDRESS;:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:MODE ADATA;NAIGNORE:
HSMODE 1;RACCESS 1;SBYTE 1;:TRIGGER:
EINTERVAL:EVENT1:I2CBUS:SBHSMODE:
TYPE HSMODE
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa?

Function Queries all settings related to the address of the I²C bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA?
-> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:
ADATA:BIT10ADDRESS:
PATTERN "10111011111";:TRIGGER:
EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN "11011110";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
TYPE BIT10ADDRESS
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT10address?

Function Queries all settings related to the 10-bit address of the I^2C bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT10address?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT10ADDRESS? -> :TRIGGER:EINTERVAL:
EVENT1:I2CBUS:ADATA:BIT10ADDRESS:
PATTERN "10111011111"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT10address:HEXA

Function Sets the 10-bit address of the I^2C bus trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT10address:HEXA {<String>}
```
<x> = 1 or 2
<String> = 3 characters by combining '0' to 'F' and 'X' (bit 8 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT10ADDRESS:HEXA "7AB"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT10address:PATTern

Function Sets the 10-bit address of the I^2C bus trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT10address:PATTern {<String>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT10address:PATTern?
```
<x> = 1 or 2
<String> = 11 characters by combining '0', '1', and 'X' (bit 8 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT10ADDRESS:PATTERN "10111011111"
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT10ADDRESS:PATTERN? -> :TRIGGER:
EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT10ADDRESS:PATTERN "10111011111"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress?

Function Queries all settings related to the 7-bit address of the I^2C bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT7ADdress?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7ADDRESS? -> :TRIGGER:EINTERVAL:
EVENT1:I2CBUS:ADATA:BIT7ADDRESS:
PATTERN "11011110"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress:HEXA

Function Sets the 7-bit address of the I^2C bus trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT7ADdress:HEXA {<String>}
```
<x> = 1 or 2
<String> = 2 characters by combining '0' to 'F' and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7ADDRESS:HEXA "DE"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress:PATTern

Function Sets the 7-bit address of the I^2C bus trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT7ADdress:PATTern {<String>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT7ADdress:PATTern?
```
<x> = 1 or 2
<String> = 8 characters by combining '0', '1', and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN "11011110"
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN? -> :TRIGGER:
EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN "11011110"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub?

Function Queries all settings related to the 7-bit + Sub address of the I^2C bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT7APsub?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
ADATA:BIT7APSUB? -> :TRIGGER:EINTERVAL:
EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS:
PATTERN "10101011";:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:ADATA:BIT7APSUB:SADDRESS:
PATTERN "10101011"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess?

Function Queries all settings related to the 7-bit address of the 7-bit + Sub address of the I^2C bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT7APsub:ADDRess?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:ADDRESS? -> :TRIGGER:
EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess:HEXA

Function Sets the 7-bit address of the 7-bit + Sub address of the I^2C bus trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT7APsub:ADDRess:HEXA {<String>}
```
<x> = 1 or 2
<String> = 2 characters by combining '0' to 'F' and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
ADATA:BIT7APSUB:ADDRESS:HEXA "AB"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess:PATTern

Function Sets the 7-bit address of the 7-bit + Sub address of the I^2C bus trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT7APsub:ADDRess:PATTern
{<String>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT7APsub:ADDRess:PATTern?
```
<x> = 1 or 2
<String> = 8 characters by combining '0', '1', and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011"
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress?

Function Queries all settings related to the Sub address of the 7-bit + Sub address of the I^2C bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT7APsub:SADDress?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:SADDRESS? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress:HEXA

Function Sets the Sub address of the 7-bit + Sub address of the I^2C bus trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT7APsub:SADDress:HEXA
{<String>}
```
<x> = 1 or 2
<String> = 2 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:HEXA "EF"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress:PATTern

Function Sets the Sub address of the 7-bit + Sub address of the I^2C bus trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT7APsub:SADDress:PATTern
{<String>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT7APsub:SADDress:PATTern?
```
<x> = 1 or 2
<String> = 8 characters by combining '0,' '1,' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011"
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:TYPE

Function Sets the address type of the I^2C bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:TYPE {BIT10address|BIT7ADdress|
BIT7APsub}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:TYPE?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
ADATA:TYPE BIT10ADDRESS
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
TYPE? -> :TRIGGER:EINTERVAL:EVENT1:
I2CBUS:ADATA:TYPE BIT10ADDRESS
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:CLOCk?

Function Queries all settings related to the clock of the I²C bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
CLOCk?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:CLOCK?
-> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:
CLOCK:SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:CLOCk:SOURce

Function Sets the clock trace of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
CLOCk:SOURce {<NRf>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
CLOCk:SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:CLOCK:
SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:CLOCK:
SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:
I2CBUS:CLOCK:SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA?

Function Queries all settings related to the data of the I²C bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
DATA? -> :TRIGGER:EINTERVAL:EVENT1:
I2CBUS:DATA:BYTE 1;CONDITION TRUE;
DPOSITION 1;MODE 1;PATTERN1 "10101011";
PATTERN2 "10101010";
PATTERN3 "10101111";
PATTERN4 "10101011";PMODE DONTCARE;
SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:BYTE

Function Sets the number of data bytes of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:BYTE {<NRf>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:BYTE?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
BYTE 1
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
BYTE? -> :TRIGGER:EINTERVAL:EVENT1:
I2CBUS:DATA:BYTE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:CONDition

Function Sets the determination method (match or not match) of the data of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:CONDition {FALSe|TRUE}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:CONDition?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
CONDITION TRUE
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
CONDITION? -> :TRIGGER:EINTERVAL:
EVENT1:I2CBUS:DATA:CONDITION TRUE
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:DPOSition

Function Sets the position for comparing the data pattern of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:DPOSition {<NRf>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:DPOSition?
```
<x> = 1 or 2
<NRf> = 0 to 9999

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
DPOSITION 1
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
DPOSITION? -> :TRIGGER:EINTERVAL:
EVENT1:I2CBUS:DATA:DPOSITION 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:HEXA<x>

Function Sets the data of the I²C bus trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:HEXA<x> {<String>}
```
<x> of EVENt<x> = 1 or 2
<x> of HEXA<x> = 1 to 4
<String> = 2 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:HEXA1 "AB"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:MODE

Function Enables/Disables the data conditions of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:MODE {<Boolean>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:MODE?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:MODE ON
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:MODE? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:MODE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:PATTern<x>

Function Sets the data of the I²C bus trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:PATTern<x> {<String>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:PATTern<x>?
```
<x> of EVENt<x> = 1 or 2
<x> of PATTern<x> = 1 to 4
<String> = 8 characters by combining '0,' '1,' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:PATTERN1 "10101011"
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:PATTERN1? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:PATTERN1 "10101011"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:PMODe

Function Sets the pattern comparison start position mode of the data of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:PMODe {DONTcare|SELect}
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:PMODe?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:PMODE SELECT
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:PMODE? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:PMODE SELECT
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:SOURce

Function Sets the data trace of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:SOURce {<NRf>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl?

Function Queries all settings related to the general call of the I²C bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:BIT7MADDRESS:PATTERN "1010101";:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:SBYTE BIT7MADDRESS
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:BIT7maddress?

Function Queries all settings related to the 7-bit master address of the general call of the I²C bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
GCALl:BIT7maddress?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
BIT7MADDRESS? -> :TRIGGER:EINTERVAL:
EVENT1:I2CBUS:GCALL:BIT7MADDRESS:
PATTERN "1010101"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:BIT7maddress:HEXA

Function Sets the 7-bit master address of the general call of the I²C bus trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
GCALl:BIT7maddress:HEXA {<String>}
```
<x> = 1 or 2
<String> = 2 characters by combining '0' to 'F' and 'X' (bit 0 is fixed 1)

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
BIT7MADDRESS:HEXA "AB"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:BIT7maddress:PATTern

Function Sets the 7-bit master address of the general call of the I²C bus trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
GCALl:BIT7maddress:PATTern {<String>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
GCALl:BIT7maddress:PATTern?
```
<x> = 1 or 2
<String> = 7 characters by combining '0,' '1,' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
BIT7MADDRESS:PATTERN "1010101"
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
BIT7MADDRESS:PATTERN? -> :TRIGGER:
EINTERVAL:EVENT1:I2CBUS:GCALL:
BIT7MADDRESS:PATTERN "1010101"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:SBYTe (Second Byte)

Function Sets the second byte type of the general call of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
GCALl:SBYTe {BIT7maddress|DONTcare|H04|
H06}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
GCALl:SBYTe?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
SBYTE BIT7MADDRESS
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
SBYTE? -> :TRIGGER:EINTERVAL:EVENT1:
I2CBUS:GCALL:SBYTE BIT7MADDRESS
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:MODE

Function Sets the trigger mode of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
MODE {ADATa|ESTart|GCALl|NAIGnore|
SBHSmode}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
MODE?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
MODE ADATA
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:MODE?
-> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:
MODE ADATA
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:NAIGnore?

Function Queries all settings related to the NON ACK ignore mode of the I²C bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
NAIGnore?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE? -> :TRIGGER:EINTERVAL:EVENT1:
I2CBUS:NAIGNORE:HSMODE 1;RACCESS 1;
SBYTE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus: NAIGnore:HSMode

Function Sets whether to ignore NON ACK in high speed mode of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
NAIGnore:HSMode {<Boolean>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
NAIGnore:HSMode?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:HSMODE ON
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:HSMODE? -> :TRIGGER:EINTERVAL:
EVENT1:I2CBUS:NAIGNORE:HSMODE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus: NAIGnore:RACCess

Function Sets whether to ignore NON ACK in read access mode of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
NAIGnore:RACCess {<Boolean>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
NAIGnore:RACCess?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:RACCESS ON
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:RACCESS? ->
:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:NAIGNORE:RACCESS 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus: NAIGnore:SBYTe (Start Byte)

Function Sets whether to ignore NON ACK in the start byte of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
NAIGnore:SBYTe {<Boolean>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
NAIGnore:SBYTe?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:SBYTE ON
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:SBYTE? -> :TRIGGER:EINTERVAL:
EVENT1:I2CBUS:NAIGNORE:SBYTE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus: SBHSmode?

Function Queries all settings related to the start byte and high speed mode of the I²C bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
SBHSmode?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
SBHSMODE? -> :TRIGGER:EINTERVAL:EVENT1:
I2CBUS:SBHSMODE:TYPE HSMODE
```

:TRIGger:EINTerval:EVENt<x>:I2CBus: SBHSmode:TYPE

Function Sets the type of the start byte or high speed mode of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
SBHSmode:TYPE {HSMode|SBYTe}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
SBHSmode:TYPE?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
SBHSMODE:TYPE HSMODE
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
SBHSMODE:TYPE? -> :TRIGGER:EINTERVAL:
EVENT1:I2CBUS:SBHSMODE:TYPE HSMODE
```

:TRIGger:EINTerval:EVENt<x>:SPATtern? (Serial Pattern)

Function Queries all settings related to the serial pattern trigger of the event.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN? ->
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:
BITRATE 1.000E+00;CLOCK:MODE 1;
POLARITY FALL;SOURCE 1;:TRIGGER:
EINTERVAL:EVENT1:SPATTERN:CS 1;DATA:
ACTIVE HIGH;SOURCE 1;:TRIGGER:
EINTERVAL:EVENT1:SPATTERN:LATCH:
SOURCE 1;POLARITY FALL;:TRIGGER:
EINTERVAL:EVENT1:SPATTERN:
PATTERN "1100110111101111"
```

:TRIGger:EINTerval:EVENt<x>:SPATtern:BITRate

Function Sets the bit rate of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:BITRate {<NRf>}
:TRIGger:EINTerval:EVENt<x>:SPATtern:BITRate?
```
<x> = 1 or 2
<NRf> = 1 to 50M (bps)

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:BITRATE 1
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:BITRATE? -> :TRIGGER:EINTERVAL:EVENT1:SPATTERN:BITRATE 1.000E+00
```

Description This command is valid when
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:MODE OFF.
```

:TRIGger:EINTerval:EVENt<x>:SPATtern:CLEar

Function Clears the entire pattern of the serial pattern trigger (to don't care).

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLEar
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLEAR
```

:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk?

Function Queries all settings related to clock of the serial pattern trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLOCK? -> :TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLOCK:MODE 1;POLARITY FALL;SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:MODE

Function Enables/Disables the clock of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:MODE {<Boolean>}
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:MODE?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLOCK:MODE ON
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLOCK:MODE? -> :TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLOCK:MODE 1
```

:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:POLarity

Function Sets the polarity of the clock trace of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:POLarity {FALL|RISE}
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:POLarity?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLOCK:POLARITY FALL
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLOCK:POLARITY? -> :TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLOCK:POLARITY FALL
```

Description This command is valid when
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:MODE ON.
```

:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:SOURce

Function Sets the clock trace of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:SOURce {<NRf>}
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLOCK:SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLOCK:SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:SPATTERN:CLOCK:SOURCE 1
```

Description This command is valid when
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:MODE ON.
```

:TRIGger:EINTerval:EVENt<x>:SPATtern:CS

Function Enables/Disables the chip select of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:CS {<Boolean>}
:TRIGger:EINTerval:EVENt<x>:SPATtern:CS?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:CS ON
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:CS? -> :TRIGGER:EINTERVAL:EVENT1:SPATTERN:CS 1
```

Description This command is valid when
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:MODE ON.
```

:TRIGger:EINTerval:EVENt<x>:SPATtern:DATA?

Function Queries all settings related to data of the serial pattern trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:
DATA?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:
DATA? -> :TRIGGER:EINTERVAL:EVENT1:
SPATTERN:DATA:ACTIVE HIGH;SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:SPATtern:DATA:ACTive

Function Sets the active level of the data of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:
DATA:ACTive {HIGH|LOW}
:TRIGger:EINTerval:EVENt<x>:SPATtern:
DATA:ACTive?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:
DATA:ACTIVE HIGH
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:
DATA:ACTIVE? -> :TRIGGER:EINTERVAL:
EVENT1:SPATTERN:DATA:ACTIVE HIGH
```

:TRIGger:EINTerval:EVENt<x>:SPATtern:DATA:SOURce

Function Sets the data trace of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:
DATA:SOURce {<NRf>}
:TRIGger:EINTerval:EVENt<x>:SPATtern:
DATA:SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:
DATA:SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:
DATA:SOURCE? -> :TRIGGER:EINTERVAL:
EVENT1:SPATTERN:DATA:SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:SPATtern:HEXA

Function Sets the pattern of the serial pattern trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:
HEXA {<String>}
```
<x> = 1 or 2
<String> = Up to 32 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:
HEXA "ABCD"
```

:TRIGger:EINTerval:EVENt<x>:SPATtern:LATCh?

Function Queries all settings related to latch of the serial pattern trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:
LATCh?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:
LATCH? -> :TRIGGER:EINTERVAL:EVENT1:
SPATTERN:LATCH:SOURCE 1;POLARITY FALL
```

:TRIGger:EINTerval:EVENt<x>:SPATtern:LATCh:POLarity

Function Sets the polarity of the latch trace of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:
LATCh:POLarity {FALL|RISE}
:TRIGger:EINTerval:EVENt<x>:SPATtern:
LATCh:POLarity?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:
LATCH:POLARITY FALL
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:
LATCH:POLARITY? -> :TRIGGER:EINTERVAL:
EVENT1:SPATTERN:LATCH:POLARITY FALL
```

Description
- This command is valid when `:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:MODE ON`.
- This command in invalid when `:TRIGger:EINTerval:EVENt<x>:SPATtern:LATCh:SOURce NONE`.

:TRIGger:EINTerval:EVENt<x>:SPATtern:LATCh:SOURce

Function Sets the latch trace of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:
LATCh:SOURce {<NRf>|NONE}
:TRIGger:EINTerval:EVENt<x>:SPATtern:
LATCh:SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:
LATCH:SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:
LATCH:SOURCE? -> :TRIGGER:EINTERVAL:
EVENT1:SPATTERN:LATCH:SOURCE 1
```

Description This command is valid when `:TRIGger:EINTerval:EVENt<x>:SPATtern:CLOCk:MODE ON`.

:TRIGger:EINTerval:EVENt<x>:SPATtern:PATTern

Function Sets the pattern of the serial pattern trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPATtern:
PATTern {<String>}
:TRIGger:EINTerval:EVENt<x>:SPATtern:
PATTern?
```
<x> = 1 or 2
<String> = Up to 128 characters by combining '0', '1', and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:
PATTERN "1100110111101111"
:TRIGGER:EINTERVAL:EVENT1:SPATTERN:
PATTERN? -> :TRIGGER:EINTERVAL:EVENT1:
SPATTERN:PATTERN "1100110111101111"
```

:TRIGger:EINTerval:EVENt<x>:SPIBus?

Function Queries all settings related to the SPI bus trigger of the event.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:
BITORDER LSBFIRST;CLOCK:POLARITY FALL;
SOURCE 1;:TRIGGER:EINTERVAL:EVENT1:
SPIBUS:CS:ACTIVE HIGH;SOURCE 1;:
TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:
BYTE 1;CONDITION TRUE;DPOSITION 1;
PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "00010010";SOURCE 3;:TRIGGER:
EINTERVAL:EVENT1:SPIBUS:DATA2:BYTE 4;
CONDITION TRUE;DPOSITION 1;
PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "00010010";SOURCE 3;:TRIGGER:
EINTERVAL:EVENT1:SPIBUS:MODE WIRE3
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:BITorder

Function Sets the bit order of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:
BITorder {LSBFirst|MSBFirst}
:TRIGger:EINTerval:EVENt<x>:SPIBus:
BITorder?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:
BITORDER LSBFIRST
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:
BITORDER? -> :TRIGGER:EINTERVAL:
EVENT1:SPIBUS:BITORDER LSBFIRST
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:CLOCk?

Function Queries all settings related to the clock of the SPI bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:
CLOCk?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CLOCK?
-> :TRIGGER:EINTERVAL:EVENT1:SPIBUS:
CLOCK:POLARITY FALL;SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:CLOCk:POLarity

Function Sets the polarity of the clock trace of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:
CLOCk:POLarity {FALL|RISE}
:TRIGger:EINTerval:EVENt<x>:SPIBus:
CLOCk:POLarity?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CLOCK:
POLARITY FALL
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CLOCK:
POLARITY? -> :TRIGGER:EINTERVAL:EVENT1:
SPIBUS:CLOCK:POLARITY FALL
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:CLOCk:SOURce

Function Sets the clock trace of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:
CLOCk:SOURce {<NRf>}
:TRIGger:EINTerval:EVENt<x>:SPIBus:
CLOCk:SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CLOCK:
SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CLOCK:
SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:
SPIBUS:CLOCK:SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:CS?

Function Queries all settings related to the chip select of the SPI bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:CS?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS:
ACTIVE HIGH;SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:CS:ACTive

Function Sets the active level of the chip select of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:CS:
ACTive {HIGH|LOW}
:TRIGger:EINTerval:EVENt<x>:SPIBus:CS:
ACTive?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS:
ACTIVE HIGH
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS:
ACTIVE? -> :TRIGGER:EINTERVAL:EVENT1:
SPIBUS:CS:ACTIVE HIGH
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:CS:SOURce

Function Sets the chip select trace of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:CS:
SOURce {<NRf>}
:TRIGger:EINTerval:EVENt<x>:SPIBus:CS:
SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS:
SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS:
SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:
SPIBUS:CS:SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>?

Function Queries all settings related to the data of the SPI bus trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:
DATA<x>?
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1?
-> :TRIGGER:EINTERVAL:EVENT1:SPIBUS:
DATA1:BYTE 1;CONDITION TRUE;
DPOSITION 1;PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "00010010";SOURCE 3
```

Description DATA2 is valid when
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:MODE
WIRE4.
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:BYTE

Function Sets the number of bytes of the data of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:
DATA<x>:BYTE {<NRf>}
:TRIGger:EINTerval:EVENt<x>:SPIBus:
DATA<x>:BYTE?
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:
BYTE 1
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:
BYTE? -> :TRIGGER:EINTERVAL:EVENT1:
SPIBUS:DATA1:BYTE 1
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:CONDition

Function Sets the determination method (match or not match) of the data of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:
DATA<x>:CONDition {FALSe|TRUE}
:TRIGger:EINTerval:EVENt<x>:SPIBus:
DATA<x>:CONDition?
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:
CONDITION TRUE
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:
CONDITION? -> :TRIGGER:EINTERVAL:
EVENT1:SPIBUS:DATA1:CONDITION TRUE
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:DPOSition

Function Sets the pattern comparison start position of the data of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:
DATA<x>:DPOSition {<NRf>}
:TRIGger:EINTerval:EVENt<x>:SPIBus:
DATA<x>:DPOSition?
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<NRf> = 0 to 9999

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:
DPOSITION 1
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:
DPOSITION? -> :TRIGGER:EINTERVAL:
EVENT1:SPIBUS:DATA1:DPOSITION 1
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:HEXA<x>

Function Sets the data of the SPI bus trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:HEXA<x> {<String>}
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<x> of HEXA<x> = 1 to 4
<String> = 2 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:HEXA1 "AB"
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:PATTern<x>

Function Sets the data of the SPI bus trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:PATTern<x> {<String>}

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:PATTern<x>?
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<x> of PATTern<x> = 1 to 4
<String> = 8 characters by combining '0,' '1,' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:PATTERN1 "10101011"
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:PATTERN1? -> :TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:PATTERN1 "10101011"
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:SOURce

Function Sets the trace of the data of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:SOURce {<NRf>}
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:SOURce?
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:MODE

Function Sets the wiring system of the SPI bus trigger (three-wire or four-wire) or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:MODE {WIRE3|WIRE4}
:TRIGger:EINTerval:EVENt<x>:SPIBus:MODE?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:MODE WIRE3
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:MODE? -> :TRIGGER:EINTERVAL:EVENT1:SPIBUS:MODE WIRE3
```

:TRIGger:EINTerval:EVENt<x>:STATe?

Function Queries all settings related to condition to be satisfied of the event.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:STATe?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:STATE? -> :TRIGGER:EINTERVAL:EVENT1:STATE:CHANNEL1 DONTCARE;CHANNEL2 DONTCARE;CHANNEL3 DONTCARE;CHANNEL4 DONTCARE;LOGIC AND
```

:TRIGger:EINTerval:EVENt<x>:STATe:CHANnel<x>

Function Sets the condition to be satisfied of the channel or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:STATe:CHANnel<x> {DONTcare|HIGH|IN|LOW|OUT}
:TRIGger:EINTerval:EVENt<x>:STATe:CHANnel<x>?
```
<x> of EVENt<x> = 1 or 2
<x> of CHANnel<x> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:STATE:CHANNEL1 HIGH
:TRIGGER:EINTERVAL:EVENT1:STATE:CHANNEL1? -> :TRIGGER:EINTERVAL:EVENT1:STATE:CHANNEL1 HIGH
```

Description
- This command is valid when `:TRIGger:EINTerval:EVENt<x>:TYPE EQUalify|I2CBus|PQUalify|PSTAte|SPATtern|STATe`.
- `{HIGH|LOW}` is valid when `:TRIGger:EINTerval:EVENt<x>:TYPE I2CBus|SPATtern`.
- `{IN|OUT}` is valid when `:TRIGger:EINTerval:EVENt<x>:TYPE EQUalify|PQUalify|PSTAte|STATe` and `TRIGger:SOURce:CHANnel<x>:WINDow ON`.
- `{HIGH|LOW}` is valid when `:TRIGger:EINTerval:EVENt<x>:TYPE EQUalify|PQUalify|PSTAte|STATe` and `TRIGger:SOURce:CHANnel<x>:WINDow OFF`.

:TRIGger:EINTerval:EVENt<x>:STATe:LOGic

Function Sets the logic of the condition to be satisfied or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:STATe:
LOGic {AND|OR}
:TRIGger:EINTerval:EVENt<x>:STATe:
LOGic?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:STATE:
LOGIC AND
:TRIGGER:EINTERVAL:EVENT1:STATE:LOGIC?
-> :TRIGGER:EINTERVAL:EVENT1:STATE:
LOGIC AND
```

Description This command is valid when
```
:TRIGger:EINTerval:EVENt<x>:TYPE
EQUalify|I2CBus|PQUalify|PSTAte|
SPATtern|STATe.
```

:TRIGger:EINTerval:EVENt<x>:TYPE

Function Sets the trigger type of the event or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:
TYPE {EDGE|EQUalify|I2CBus|PQUalify|
PSTAte|PULSe|SPATtern|SPIBus|STATe}
:TRIGger:EINTerval:EVENt<x>:TYPE?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:TYPE EDGE
:TRIGGER:EINTERVAL:EVENT1:TYPE? ->
:TRIGGER:EINTERVAL:EVENT1:TYPE EDGE
```

:TRIGger:EINTerval:EVENt<x>:WIDTh?

Function Queries all settings related to the pulse width trigger of the event.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:WIDTh?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:WIDTH? ->
:TRIGGER:EINTERVAL:EVENT1:WIDTH:
MODE TIMEOUT;POLARITY POSITIVE;
SOURCE EXTERNAL;TIME1 1.000E+00;
TIME2 1.000E+00
```

:TRIGger:EINTerval:EVENt<x>:WIDTh:MODE

Function Sets the determination mode of the pulse width trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:WIDTh:
MODE {BETWeen|IN|NOTBetween|OUT|
TIMeout}
:TRIGger:EINTerval:EVENt<x>:WIDTh:MODE?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:WIDTH:MODE
TIMEOUT
:TRIGGER:EINTERVAL:EVENT1:WIDTH:MODE?
-> :TRIGGER:EINTERVAL:EVENT1:WIDTH:
MODE TIMEOUT
```

:TRIGger:EINTerval:EVENt<x>:WIDTh:POLarity

Function Sets the polarity of the pulse width trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:WIDTh:
POLarity {FALSe|IN|NEGative|OUT|
POSitive|TRUE}
:TRIGger:EINTerval:EVENt<x>:WIDTh:
POLarity?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:WIDTH:
POLARITY POSITIVE
:TRIGGER:EINTERVAL:EVENT1:WIDTH:
POLARITY? -> :TRIGGER:EINTERVAL:EVENT1:
WIDTH:POLARITY POSITIVE
```

Description
- {IN|OUT} is valid when :TRIGger:EINTerval:EVENt<x>:TYPE PQUalify|PULSe and :TRIGger:SOURce:CHANnel<x>:WINDow ON. {POSitive|NEGative} is valid when :TRIGger:EINTerval:EVENt<x>:TYPE PQUalify|PULSe and :TRIGger:SOURce:CHANnel<x>:WINDow OFF.
- {FALSe|TRUE} is valid when :TRIGger:EINTerval:EVENt<x>:TYPE PSTAte.

:TRIGger:EINTerval:EVENt<x>:WIDTh:SOURce

Function Sets the trigger source of the pulse width trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:WIDTh:SOURce {<NRf>|EXTernal}
:TRIGger:EINTerval:EVENt<x>:WIDTh:SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:WIDTH:SOURCE EXTERNAL
:TRIGGER:EINTERVAL:EVENT1:WIDTH:SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:WIDTH:SOURCE EXTERNAL
```

Description This command is valid when `:TRIGger:EINTerval:EVENt<x>:TYPE PQUalify|PULSe`.

:TRIGger:EINTerval:EVENt<x>:WIDTh:TIME<x>

Function Sets the pulse width of the pulse width trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:WIDTh:TIME<x> {<Time>}
:TRIGger:EINTerval:EVENt<x>:WIDTh:TIME<x>?
```
<x> of EVENt<x> = 1 or 2
<x> of TIME<x> = 1 or 2
<Time> = 1 ns to 10 s (500 ps steps)

Example
```
:TRIGGER:EINTERVAL:EVENT1:WIDTH:TIME1 1S
:TRIGGER:EINTERVAL:EVENT1:WIDTH:TIME1? -> :TRIGGER:EINTERVAL:EVENT1:WIDTH:TIME1 1.000E+00
```

Description TIME2 is valid when `:TRIGger:EINTerval:EVENt<x>:WIDTh:MODE BETWeen|NOTBetween`.

:TRIGger:EINTerval:MODE

Function Sets the determination mode of the event interval or queries the current setting.

Syntax
```
:TRIGger:EINTerval:MODE {BETWeen|IN|NOTBetween|OUT|TIMeout}
:TRIGger:EINTerval:MODE?
```

Example
```
:TRIGGER:EINTERVAL:MODE BETWEEN
:TRIGGER:EINTERVAL:MODE? -> :TRIGGER:EINTERVAL:MODE BETWEEN
```

:TRIGger:EINTerval:TIME<x>

Function Sets the interval time of the event interval or queries the current setting.

Syntax
```
:TRIGger:EINTerval:TIME<x> {<Time>}
:TRIGger:EINTerval:TIME<x>?
```
<x> = 1 or 2
<Time> = 1 ns to 10 s (500 ps steps)

Example
```
:TRIGGER:EINTERVAL:TIME1 1S
:TRIGGER:EINTERVAL:TIME1? -> :TRIGGER:EINTERVAL:TIME1 0.000E+00
```

Description TIME2 is valid when `:TRIGger:EINTerval:MODE BETWeen|NOTBetween`.

:TRIGger:EINTerval:TRY?

Function Queries all settings related to the event interval trial.

Syntax
```
:TRIGger:EINTerval:TRY?
```

Example
```
:TRIGGER:EINTERVAL:TRY? -> :TRIGGER:EINTERVAL:TRY:MODE 0;SELECT 1
```

:TRIGger:EINTerval:TRY:MODE

Function Sets the trial mode or queries the current setting.

Syntax
```
:TRIGger:EINTerval:TRY:MODE {<Boolean>}
:TRIGger:EINTerval:TRY:MODE?
```

Example
```
:TRIGGER:EINTERVAL:TRY:MODE ON
:TRIGGER:EINTERVAL:TRY:MODE? -> :TRIGGER:EINTERVAL:TRY:MODE 1
```

Description This command is valid when `:TRIGger:EINTerval:MODE BETWeen|NOTBetween`.

:TRIGger:EINTerval:TRY:SELect

Function Sets the source event of the trial mode or queries the current setting.

Syntax
```
:TRIGger:EINTerval:TRY:SELect {<NRf>}
:TRIGger:EINTerval:TRY:SELect?
```
<NRf> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:TRY:SELECT 1
:TRIGGER:EINTERVAL:TRY:SELECT? -> :TRIGGER:EINTERVAL:TRY:SELECT 1
```

Description This command is valid when `:TRIGger:EINTerval:MODE BETWeen|NOTBetween`.

:TRIGger:ENHanced?

Function Queries all settings related to the enhanced trigger.

Syntax :TRIGger:ENHanced?

Example
```
:TRIGGER:ENHANCED? -> :TRIGGER:
ENHANCED:I2CBUS:ADATA:BIT10ADDRESS:
PATTERN "00011111101";:TRIGGER:
ENHANCED:I2CBUS:ADATA:BIT7ADDRESS:
PATTERN "11011110";:TRIGGER:ENHANCED:
I2CBUS:ADATA:BIT7APSUB:ADDRESS:
PATTERN "11001101";:TRIGGER:ENHANCED:
I2CBUS:ADATA:BIT7APSUB:SADDRESS:
PATTERN  "11101111";:TRIGGER:ENHANCED:
I2CBUS:ADATA:TYPE BIT10ADDRESS;:
TRIGGER:ENHANCED:I2CBUS:CLOCK:
SOURCE 1;:TRIGGER:ENHANCED:I2CBUS:DATA:
BYTE 1;CONDITION TRUE;DPOSITION 1;
MODE 1;PATTERN1 "10101011";
PATTERN2 "10001011";
PATTERN3 "00101011";
PATTERN4 "10101011";PMODE DONTCARE;
SOURCE 1;:TRIGGER:ENHANCED:I2CBUS:
GCALL:BIT7MADDRESS:PATTERN "1010101";:
TRIGGER:ENHANCED:I2CBUS:GCALL:
SBYTE BIT7MADDRESS;:TRIGGER:ENHANCED:
I2CBUS:MODE ADATA;NAIGNORE:HSMODE 1;
RACCESS 1;SBYTE 1;:TRIGGER:ENHANCED:
I2CBUS:SBHSMODE:TYPE HSMODE;:TRIGGER:
ENHANCED:SPATTERN:BITRATE 1.000E+00;
CLOCk:MODE 1;POLARITY FALL;SOURCE 1;:
TRIGGER:ENHANCED:SPATTERN:CS 1;DATA:
ACTIVE HIGH;SOURCE 1;:TRIGGER:ENHANCED:
SPATTERN:LATCH:SOURCE 1;POLARITY FALL;:
TRIGGER:ENHANCED:SPATTERN:
PATTERN "1100110111101111";:TRIGGER:
ENHANCED:SPIBUS:BITORDER MSBFIRST;
CLOCK:POLARITY FALL;SOURCE 1;:TRIGGER:
ENHANCED:SPIBUS:CS:ACTIVE HIGH;
SOURCE 1;:TRIGGER:ENHANCED:SPIBUS:
DATA1:BYTE 1;CONDITION TRUE;
DPOSITION 0;PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "00010010";SOURCE 3;:
TRIGGER:ENHANCED:SPIBUS:DATA2:BYTE 1;
CONDITION TRUE;DPOSITION 0;
PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "00010010";SOURCE DONTCARE;:
TRIGGER:ENHANCED:SPIBUS:MODE WIRE3.....
```

:TRIGger:ENHanced:I2CBus?

Function Queries all settings related to the I^2C trigger.

Syntax :TRIGger:ENHanced:I2CBus?

Example
```
:TRIGGER:ENHANCED:I2CBUS? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT10ADDRESS:PATTERN "10111011111";:
TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN "11011110";:
TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011";:
TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011";:
TRIGGER:ENHANCED:I2CBUS:ADATA:
TYPE BIT10ADDRESS;:TRIGGER:ENHANCED:
I2CBUS:CLOCK:SOURCE 1;:TRIGGER:
ENHANCED:I2CBUS:DATA:BYTE 1;
CONDITION TRUE;DPOSITION 1;MODE 1;
PATTERN1 "10101011";
PATTERN2 "10101010";
PATTERN3 "10101111";
PATTERN4 "10101011";PMODE DONTCARE;
SOURCE 1;:TRIGGER:ENHANCED:I2CBUS:
GCALL:BIT7MADDRESS:PATTERN "1010101";:
TRIGGER:ENHANCED:I2CBUS:GCALL:
SBYTE BIT7MADDRESS;:TRIGGER:ENHANCED:
I2CBUS:MODE ADATA;NAIGNORE:HSMODE 1;
RACCESS 1;SBYTE 1;:TRIGGER:ENHANCED:
I2CBUS:SBHSMODE:TYPE HSMODE
```

:TRIGger:ENHanced:I2CBus:ADATa?

Function Queries all settings related to the address of the I^2C bus trigger.

Syntax :TRIGger:ENHanced:I2CBus:ADATa?

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT10ADDRESS:PATTERN "10111011111";:
TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN "11011110";:
TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011";:
TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011";:
TRIGGER:ENHANCED:I2CBUS:ADATA:
TYPE BIT10ADDRESS
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT10address?

Function Queries all settings related to the 10-bit address of the I^2C bus trigger.

Syntax :TRIGger:ENHanced:I2CBus:ADATa:BIT10address?

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT10ADDRESS? -> :TRIGGER:ENHANCED:
I2CBUS:ADATA:BIT10ADDRESS:
PATTERN "10111011111"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT10address:HEXA

Function Sets the 10-bit address of the I²C bus trigger in hexadecimal notation.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT10address:HEXA {<String>}
```
<String> = 3 characters by combining '0' to 'F' and 'X' (bit 8 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT10ADDRESS:HEXA "7AB"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT10address:PATTern

Function Sets the 10-bit address of the I²C bus trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT10address:PATTern {<String>}
:TRIGger:ENHanced:I2CBus:ADATa:
BIT10address:PATTern?
```
<String> = 11 characters by combining '0', '1', and 'X' (bit 8 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT10ADDRESS:PATTERN "10111011111"
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT10ADDRESS:PATTERN? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT10ADDRESS:PATTERN "10111011111"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7ADdress?

Function Queries all settings related to the 7-bit address of the I²C bus trigger.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7ADdress?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7ADDRESS? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN "11011110"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7ADdress:HEXA

Function Sets the 7-bit address of the I²C bus trigger in hexadecimal notation.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7ADdress:HEXA {<String>}
```
<String> = 2 characters by combining '0' to 'F' and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7ADDRESS:HEXA "DE"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7ADdress:PATTern

Function Sets the 7-bit address of the I²C bus trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7ADdress:PATTern {<String>}
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7ADdress:PATTern?
```
<String> = 8 characters by combining '0', '1', and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN "11011110"
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN "11011110"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub?

Function Queries all settings related to the 7-bit + Sub address of the I²C bus trigger.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB? -> :TRIGGER:ENHANCED:I2CBUS:
ADATA:BIT7APSUB:ADDRESS:
PATTERN "10101011";:TRIGGER:ENHANCED:
I2CBUS:ADATA:BIT7APSUB:SADDRESS:
PATTERN "10101011"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:ADDRess?

Function Queries all settings related to the 7-bit address of the 7-bit + Sub address of the I²C bus trigger.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:ADDRess?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:ADDRess:HEXA

Function Sets the 7-bit address of the 7-bit + Sub address of the I²C bus trigger in hexadecimal notation.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:ADDRess:HEXA {String>}
```
<String> = 2 characters by combining '0' to 'F' and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:HEXA "AB"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:ADDRess:PATTern

Function Sets the 7-bit address of the 7-bit + Sub address of the I²C bus trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:ADDRess:PATTern {<String>}
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:ADDRess:PATTern?
```
<String> = 8 characters by combining '0', '1', and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011"
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:SADDress?

Function Queries all settings related to the Sub address of the 7-bit + Sub address of the I²C bus trigger.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:SADDress?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:SADDress:HEXA

Function Sets the Sub address of the 7-bit + Sub address of the I²C bus trigger in hexadecimal notation.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:SADDress:HEXA {<String>}
```
<String> = 2 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:HEXA "EF"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:SADDress:PATTern

Function Sets the Sub address of the 7-bit + Sub address of the I²C bus trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:SADDress:PATTern {<String>}
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:SADDress:PATTern?
```
<String> = 8 characters by combining '0,' '1,' and 'X'

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011"
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011"
```

:TRIGger:ENHanced:I2CBus:ADATa:TYPE

Function Sets the address type of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
TYPE {BIT10address|BIT7ADdress|
BIT7APsub}
:TRIGger:ENHanced:I2CBus:ADATa:TYPE?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
TYPE BIT10ADDRESS
:TRIGGER:ENHANCED:I2CBUS:ADATA:TYPE? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
TYPE BIT10ADDRESS
```

:TRIGger:ENHanced:I2CBus:CLOCk?

Function Queries all settings related to the clock of the I²C bus trigger.

Syntax
```
:TRIGger:ENHanced:I2CBus:CLOCk?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:CLOCK? ->
:TRIGGER:ENHANCED:I2CBUS:CLOCK:SOURCE 1
```

:TRIGger:ENHanced:I2CBus:CLOCk:SOURce

Function Sets the clock trace of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:CLOCk:
SOURce {<NRf>}
:TRIGger:ENHanced:I2CBus:CLOCk:SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:I2CBUS:CLOCK:SOURCE 1
:TRIGGER:ENHANCED:I2CBUS:CLOCK:SOURCE?
-> :TRIGGER:ENHANCED:I2CBUS:CLOCK:
SOURCE 1
```

:TRIGger:ENHanced:I2CBus:DATA?

Function Queries all settings related to the data of the I²C bus trigger.

Syntax
```
:TRIGger:ENHanced:I2CBus:DATA?
```
<x> = 1 or 2

Example
```
:TRIGGER:ENHANCED:I2CBUS:DATA? ->
:TRIGGER:ENHANCED:I2CBUS:DATA:BYTE 1;
CONDITION TRUE;DPOSITION 1;MODE 1;
PATTERN1 "10101011";
PATTERN2 "10101010";
PATTERN3 "10101111";
PATTERN4 "10101011";PMODE DONTCARE;
SOURCE 1
```

:TRIGger:ENHanced:I2CBus:DATA:BYTE

Function Sets the number of data bytes of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:DATA:
BYTE {<NRf>}
:TRIGger:ENHanced:I2CBus:DATA:BYTE?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:I2CBUS:DATA:BYTE 1
:TRIGGER:ENHANCED:I2CBUS:DATA:BYTE? ->
:TRIGGER:ENHANCED:I2CBUS:DATA:BYTE 1
```

:TRIGger:ENHanced:I2CBus:DATA:CONDition

Function Sets the determination method (match or not match) of the data of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:DATA:
CONDition {FALSe|TRUE}
:TRIGger:ENHanced:I2CBus:DATA:
CONDition?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:DATA:
CONDITION TRUE
:TRIGGER:ENHANCED:I2CBUS:DATA:
CONDITION? -> :TRIGGER:ENHANCED:I2CBUS:
DATA:CONDITION TRUE
```

:TRIGger:ENHanced:I2CBus:DATA:DPOSition

Function Sets the position for comparing the data pattern of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:DATA:
DPOSition {<NRf>}
:TRIGger:ENHanced:I2CBus:DATA:
DPOSition?
```
<NRf> = 0 to 9999

Example
```
:TRIGGER:ENHANCED:I2CBUS:DATA:
DPOSITION 1
:TRIGGER:ENHANCED:I2CBUS:DATA:
DPOSITION? -> :TRIGGER:ENHANCED:I2CBUS:
DATA:DPOSITION 1
```

:TRIGger:ENHanced:I2CBus:DATA:HEXA<x>

Function Sets the data of the I²C bus trigger in hexadecimal notation.

Syntax
```
:TRIGger:ENHanced:I2CBus:DATA:
HEXA<x> {<String>}
```
<x> = 1 to 4
<String> = 2 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:ENHANCED:I2CBUS:DATA:
HEXA1 "AB"
```

:TRIGger:ENHanced:I2CBus:DATA:MODE

Function Enables/Disables the data conditions of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:DATA:
MODE {<Boolean>}
:TRIGger:ENHanced:I2CBus:DATA:MODE?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:DATA:MODE ON
:TRIGGER:ENHANCED:I2CBUS:DATA:MODE? ->
:TRIGGER:ENHANCED:I2CBUS:DATA:MODE 1
```

:TRIGger:ENHanced:I2CBus:DATA:PATTern<x>

Function Sets the data of the I²C bus trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:DATA:
PATTern<x> {<String>}
:TRIGger:ENHanced:I2CBus:DATA:
PATTern<x>?
```
<x> = 1 to 4
<String> = 8 characters by combining '0,' '1,' and 'X'

Example
```
:TRIGGER:ENHANCED:I2CBUS:DATA:
PATTERN1 "10101011"
:TRIGGER:ENHANCED:I2CBUS:DATA:PATTERN1?
-> :TRIGGER:ENHANCED:I2CBUS:DATA:
PATTERN1 "10101011"
```

:TRIGger:ENHanced:I2CBus:DATA:PMODe

Function Sets the pattern comparison start position mode of the data of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:DATA:
PMODe {DONTcare|SELect}
:TRIGger:ENHanced:I2CBus:DATA:PMODe?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:DATA:
PMODE SELECT
:TRIGGER:ENHANCED:I2CBUS:DATA:PMODE? ->
:TRIGGER:ENHANCED:I2CBUS:DATA:
PMODE SELECT
```

:TRIGger:ENHanced:I2CBus:DATA:SOURce

Function Sets the data trace of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:DATA:
SOURce {<NRf>}
:TRIGger:ENHanced:I2CBus:DATA:SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:I2CBUS:DATA:SOURCE 1
:TRIGGER:ENHANCED:I2CBUS:DATA:SOURCE?
-> :TRIGGER:ENHANCED:I2CBUS:DATA:
SOURCE 1
```

:TRIGger:ENHanced:I2CBus:GCALl?

Function Queries all settings related to the general call of the I²C bus trigger.

Syntax `:TRIGger:ENHanced:I2CBus:GCALl?`
<x> = 1 or 2

Example
```
:TRIGGER:ENHANCED:I2CBUS:GCALL? ->
:TRIGGER:ENHANCED:I2CBUS:GCALL:
BIT7MADDRESS:PATTERN "1010101";:
TRIGGER:ENHANCED:I2CBUS:GCALL:
SBYTE BIT7MADDRESS
```

:TRIGger:ENHanced:I2CBus:GCALl: BIT7maddress?

Function Queries all settings related to the 7-bit master address of the general call of the I²C bus trigger.

Syntax `:TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress?`
<x> = 1 or 2

Example
```
:TRIGGER:ENHANCED:I2CBUS:GCALL:
BIT7MADDRESS? -> :TRIGGER:ENHANCED:
I2CBUS:GCALL:BIT7MADDRESS:
PATTERN "1010101"
```

:TRIGger:ENHanced:I2CBus:GCALl: BIT7maddress:HEXA

Function Sets the 7-bit master address of the general call of the I²C bus trigger in hexadecimal notation.

Syntax `:TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress:HEXA {<String>}`
<x> = 1 or 2
<String> = 2 characters by combining '0' to 'F' and 'X' (bit 0 is fixed 1)

Example
```
:TRIGGER:ENHANCED:I2CBUS:GCALL:
BIT7MADDRESS:HEXA "AB"
```

:TRIGger:ENHanced:I2CBus:GCALl: BIT7maddress:PATTern

Function Sets the 7-bit master address of the general call of the I²C bus trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:GCALl:
BIT7maddress:PATTern {<String>}
:TRIGger:ENHanced:I2CBus:GCALl:
BIT7maddress:PATTern?
```
<x> = 1 or 2
<String> = 7 characters by combining '0,' '1,' and 'X'

Example
```
:TRIGGER:ENHANCED:I2CBUS:GCALL:
BIT7MADDRESS:PATTERN "1010101"
:TRIGGER:ENHANCED:I2CBUS:GCALL:
BIT7MADDRESS:PATTERN? -> :TRIGGER:
ENHANCED:I2CBUS:GCALL:BIT7MADDRESS:
PATTERN "1010101"
```

:TRIGger:ENHanced:I2CBus:GCALl:SBYTe (Second Byte)

Function Sets the second byte type of the general call of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:GCALl:
SBYTe {BIT7maddress|DONTcare|H04|H06}
:TRIGger:ENHanced:I2CBus:GCALl:SBYTe?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:GCALL:
SBYTE BIT7MADDRESS
:TRIGGER:ENHANCED:I2CBUS:GCALL:SBYTE?
-> :TRIGGER:ENHANCED:I2CBUS:GCALL:
SBYTE BIT7MADDRESS
```

:TRIGger:ENHanced:I2CBus:MODE

Function Sets the trigger mode of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:MODE {ADATa|
ESTart|GCALl|NAIGnore|SBHSmode}
:TRIGger:ENHanced:I2CBus:MODE?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:MODE ADATA
:TRIGGER:ENHANCED:I2CBUS:MODE? ->
:TRIGGER:ENHANCED:I2CBUS:MODE ADATA
```

:TRIGger:ENHanced:I2CBus:NAIGnore?

Function Queries all settings related to the NON ACK ignore mode of the I²C bus trigger.

Syntax `:TRIGger:ENHanced:I2CBus:NAIGnore?`

Example
```
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE? ->
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
HSMODE 1;RACCESS 1;SBYTE 1
```

:TRIGger:ENHanced:I2CBus:NAIGnore: HSMode

Function Sets whether to ignore NON ACK in high speed mode of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:NAIGnore:
HSMode {<Boolean>}
:TRIGger:ENHanced:I2CBus:NAIGnore:
HSMode?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
HSMODE ON
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
HSMODE? -> :TRIGGER:ENHANCED:I2CBUS:
NAIGNORE:HSMODE 1
```

:TRIGger:ENHanced:I2CBus:NAIGnore:RACCess

Function Sets whether to ignore NON ACK in read access mode of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:NAIGnore:
RACCess {<Boolean>}
:TRIGger:ENHanced:I2CBus:NAIGnore:
RACCess?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
RACCESS ON
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
RACCESS? -> :TRIGGER:ENHANCED:I2CBUS:
NAIGNORE:RACCESS 1
```

:TRIGger:ENHanced:I2CBus:NAIGnore:SBYTe (Start Byte)

Function Sets whether to ignore NON ACK in the start byte of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:NAIGnore:
SBYTe {<Boolean>}
:TRIGger:ENHanced:I2CBus:NAIGnore:
SBYTe?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
SBYTE ON
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
SBYTE? -> :TRIGGER:ENHANCED:I2CBUS:
NAIGNORE:SBYTE 1
```

:TRIGger:ENHanced:I2CBus:SBHSmode?

Function Queries all settings related to the start byte and high speed mode of the I²C bus trigger.

Syntax
```
:TRIGger:ENHanced:I2CBus:SBHSmode?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:SBHSMODE? ->
:TRIGGER:ENHANCED:I2CBUS:SBHSMODE:
TYPE HSMODE
```

:TRIGger:ENHanced:I2CBus:SBHSmode:TYPE

Function Sets the type of the start byte or high speed mode of the I²C bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:SBHSmode:
TYPE {HSMode|SBYTe}
:TRIGger:ENHanced:I2CBus:SBHSmode:TYPE?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:SBHSMODE:
TYPE HSMODE
:TRIGGER:ENHANCED:I2CBUS:SBHSMODE:TYPE?
-> :TRIGGER:ENHANCED:I2CBUS:SBHSMODE:
TYPE HSMODE
```

:TRIGger:ENHanced:SPATtern? (Serial Pattern)

Function Queries all settings related to the serial pattern trigger.

Syntax
```
:TRIGger:ENHanced:SPATtern?
```

Example
```
:TRIGGER:ENHANCED:SPATTERN? ->
:TRIGGER:ENHANCED:SPATTERN:
BITRATE 1.000E+00;CLOCK:MODE 1;
POLARITY FALL;SOURCE 1;:TRIGGER:
ENHANCED:SPATTERN:CS 1;DATA:
ACTIVE HIGH;SOURCE 1;:TRIGGER:ENHANCED:
SPATTERN:LATCH:SOURCE 1;POLARITY FALL;:
TRIGGER:ENHANCED:SPATTERN:
PATTERN "1100110111101111"
```

:TRIGger:ENHanced:SPATtern:BITRate

Function Sets the bit rate of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPATtern:BITRate {<NRf>}
:TRIGger:ENHanced:SPATtern:BITRate?
```
<NRf> = 1 to 50M (bps)

Example
```
:TRIGGER:ENHANCED:SPATTERN:BITRATE 1
:TRIGGER:ENHANCED:SPATTERN:BITRATE? ->
:TRIGGER:ENHANCED:SPATTERN:
BITRATE 1.000E+00
```

Description This command is valid when
```
:TRIGger:ENHanced:SPATtern:CLOCk:MODE
```
OFF.

:TRIGger:ENHanced:SPATtern:CLEar

Function Clears the entire pattern of the serial pattern trigger (to don't care).

Syntax
```
:TRIGger:ENHanced:SPATtern:CLEar
```

Example
```
:TRIGGER:ENHANCED:SPATTERN:CLEAR
```

:TRIGger:ENHanced:SPATtern:CLOCk?

Function Queries all settings related to clock of the serial pattern trigger.

Syntax
```
:TRIGger:ENHanced:SPATtern:CLOCk?
```

Example
```
:TRIGGER:ENHANCED:SPATTERN:CLOCK? ->
:TRIGGER:ENHANCED:SPATTERN:CLOCK:
MODE 1;POLARITY FALL;SOURCE 1
```

:TRIGger:ENHanced:SPATtern:CLOCk:MODE

Function Enables/Disables the clock of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPATtern:CLOCk:
MODE {<Boolean>}
:TRIGger:ENHanced:SPATtern:CLOCk:MODE?
```

Example
```
:TRIGGER:ENHANCED:SPATTERN:CLOCK:
MODE ON
:TRIGGER:ENHANCED:SPATTERN:CLOCK:MODE?
-> :TRIGGER:ENHANCED:SPATTERN:CLOCK:
MODE 1
```

:TRIGger:ENHanced:SPATtern:CLOCk:POLarity

Function Sets the polarity of the clock trace of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPATtern:CLOCk:POLarity {FALL|RISE}
:TRIGger:ENHanced:SPATtern:CLOCk:POLarity?
```

Example
```
:TRIGGER:ENHANCED:SPATTERN:CLOCK:POLARITY FALL
:TRIGGER:ENHANCED:SPATTERN:CLOCK:POLARITY? -> :TRIGGER:ENHANCED:SPATTERN:CLOCK:POLARITY FALL
```

Description This command is valid when
```
:TRIGger:ENHanced:SPATtern:CLOCk:MODE ON.
```

:TRIGger:ENHanced:SPATtern:CLOCk:SOURce

Function Sets the clock trace of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPATtern:CLOCk:SOURce {<NRf>}
:TRIGger:ENHanced:SPATtern:CLOCk:SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:SPATTERN:CLOCK:SOURCE 1
:TRIGGER:ENHANCED:SPATTERN:CLOCK:SOURCE? -> :TRIGGER:ENHANCED:SPATTERN:CLOCK:SOURCE 1
```

Description This command is valid when
```
:TRIGger:ENHanced:SPATtern:CLOCk:MODE ON.
```

:TRIGger:ENHanced:SPATtern:CS

Function Enables/Disables the chip select of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPATtern:CS {<Boolean>}
:TRIGger:ENHanced:SPATtern:CS?
```

Example
```
:TRIGGER:ENHANCED:SPATTERN:CS ON
:TRIGGER:ENHANCED:SPATTERN:CS? -> :TRIGGER:ENHANCED:SPATTERN:CS 1
```

Description This command is valid when
```
:TRIGger:ENHanced:SPATtern:CLOCk:MODE ON.
```

:TRIGger:ENHanced:SPATtern:DATA?

Function Queries all settings related to data of the serial pattern trigger.

Syntax
```
:TRIGger:ENHanced:SPATtern:DATA?
```

Example
```
:TRIGGER:ENHANCED:SPATTERN:DATA? -> :TRIGGER:ENHANCED:SPATTERN:DATA:ACTIVE HIGH;SOURCE 1
```

:TRIGger:ENHanced:SPATtern:DATA:ACTive

Function Sets the active level of the data of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPATtern:DATA:ACTive {HIGH|LOW}
:TRIGger:ENHanced:SPATtern:DATA:ACTive?
```

Example
```
:TRIGGER:ENHANCED:SPATTERN:DATA:ACTIVE HIGH
:TRIGGER:ENHANCED:SPATTERN:DATA:ACTIVE? -> :TRIGGER:ENHANCED:SPATTERN:DATA:ACTIVE HIGH
```

:TRIGger:ENHanced:SPATtern:DATA:SOURce

Function Sets the data trace of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPATtern:DATA:SOURce {<NRf>}
:TRIGger:ENHanced:SPATtern:DATA:SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:SPATTERN:DATA:SOURCE 1
:TRIGGER:ENHANCED:SPATTERN:DATA:SOURCE? -> :TRIGGER:ENHANCED:SPATTERN:DATA:SOURCE 1
```

:TRIGger:ENHanced:SPATtern:HEXA

Function Sets the pattern of the serial pattern trigger in hexadecimal notation.

Syntax
```
:TRIGger:ENHanced:SPATtern:HEXA {<String>}
```
<String> = Up to 32 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:ENHANCED:SPATTERN:HEXA "ABCD"
```

:TRIGger:ENHanced:SPATtern:LATCh?

Function Queries all settings related to latch of the serial pattern trigger.

Syntax
```
:TRIGger:ENHanced:SPATtern:LATCh?
```

Example
```
:TRIGGER:ENHANCED:SPATTERN:LATCH? -> :TRIGGER:ENHANCED:SPATTERN:LATCH:SOURCE 1;POLARITY FALL
```

:TRIGger:ENHanced:SPATtern:LATCh:POLarity

Function Sets the polarity of the latch trace of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPATtern:LATCh:
POLarity {FALL|RISE}
:TRIGger:ENHanced:SPATtern:LATCh:
POLarity?
```

Example
```
:TRIGGER:ENHANCED:SPATTERN:LATCH:
POLARITY FALL
:TRIGGER:ENHANCED:SPATTERN:LATCH:
POLARITY? -> :TRIGGER:ENHANCED:
SPATTERN:LATCH:POLARITY FALL
```

Description
- This command is valid when `:TRIGger:ENHanced:SPATtern:CLOCk:MODE ON`.
- This command in invalid when `:TRIGger:ENHanced:SPATtern:LATCh:SOURce NONE`.

:TRIGger:ENHanced:SPATtern:LATCh:SOURce

Function Sets the latch trace of the serial pattern trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPATtern:LATCh:
SOURce {<NRf>|NONE}
:TRIGger:ENHanced:SPATtern:LATCh:
SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:SPATTERN:LATCH:
SOURCE 1
:TRIGGER:ENHANCED:SPATTERN:LATCH:
SOURCE? -> :TRIGGER:ENHANCED:SPATTERN:
LATCH:SOURCE 1
```

Description This command is valid when `:TRIGger:ENHanced:SPATtern:CLOCk:MODE ON`.

:TRIGger:ENHanced:SPATtern:PATTern

Function Sets the pattern of the serial pattern trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPATtern:
PATTern {<String>}
:TRIGger:ENHanced:SPATtern:PATTern?
```
<String> = Up to 128 characters by combining '0', '1', and 'X'

Example
```
:TRIGGER:ENHANCED:SPATTERN:PATTERN
"1100110111101111"
:TRIGGER:ENHANCED:SPATTERN:PATTERN? ->
:TRIGGER:ENHANCED:SPATTERN:
PATTERN "1100110111101111"
```

:TRIGger:ENHanced:SPIBus?

Function Queries all settings related to the SPI bus trigger.

Syntax
```
:TRIGger:ENHanced:SPIBus?
```

Example
```
:TRIGGER:ENHANCED:SPIBUS? -> :TRIGGER:
ENHANCED:SPIBUS:BITORDER LSBFIRST;
CLOCK:POLARITY FALL;SOURCE 1;:TRIGGER:
ENHANCED:SPIBUS:CS:ACTIVE HIGH;
SOURCE 1;:TRIGGER:ENHANCED:SPIBUS:
DATA1:BYTE 1;CONDITION TRUE;
DPOSITION 1;PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "00010010";SOURCE 3;:TRIGGER:
ENHANCED:SPIBUS:DATA2:BYTE 4;
CONDITION TRUE;DPOSITION 1;
PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "00010010";SOURCE 3;:TRIGGER:
ENHANCED:SPIBUS:MODE WIRE3
```

:TRIGger:ENHanced:SPIBus:BITorder

Function Sets the bit order of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:
BITorder {LSBFirst|MSBFirst}
:TRIGger:ENHanced:SPIBus:BITorder?
```

Example
```
:TRIGGER:ENHANCED:SPIBUS:
BITORDER LSBFIRST
:TRIGGER:ENHANCED:SPIBUS:BITORDER? ->
:TRIGGER:ENHANCED:SPIBUS:
BITORDER LSBFIRST
```

:TRIGger:ENHanced:SPIBus:CLOCk?

Function Queries all settings related to the clock of the SPI bus trigger.

Syntax
```
:TRIGger:ENHanced:SPIBus:CLOCk?
```

Example
```
:TRIGGER:ENHANCED:SPIBUS:CLOCK? ->
:TRIGGER:ENHANCED:SPIBUS:CLOCK:
POLARITY FALL;SOURCE 1
```

:TRIGger:ENHanced:SPIBus:CLOCk:POLarity

Function Sets the polarity of the clock trace of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:CLOCk:
POLarity {FALL|RISE}
:TRIGger:ENHanced:SPIBus:CLOCk:
POLarity?
```

Example
```
:TRIGGER:ENHANCED:SPIBUS:CLOCK:
POLARITY FALL
:TRIGGER:ENHANCED:SPIBUS:CLOCK:
POLARITY? -> :TRIGGER:ENHANCED:SPIBUS:
CLOCK:POLARITY FALL
```

:TRIGger:ENHanced:SPIBus:CLOCk:SOURce

Function Sets the clock trace of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:CLOCk:
SOURce {<NRf>}
:TRIGger:ENHanced:SPIBus:CLOCk:SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:SPIBUS:CLOCK:SOURCE 1
:TRIGGER:ENHANCED:SPIBUS:CLOCK:SOURCE?
-> :TRIGGER:ENHANCED:SPIBUS:CLOCK:
SOURCE 1
```

:TRIGger:ENHanced:SPIBus:CS?

Function Queries all settings related to the chip select of the SPI bus trigger.

Syntax
```
:TRIGger:ENHanced:SPIBus:CS?
```

Example
```
:TRIGGER:ENHANCED:SPIBUS:CS? ->
:TRIGGER:ENHANCED:SPIBUS:CS:
ACTIVE HIGH;SOURCE 1
```

:TRIGger:ENHanced:SPIBus:CS:ACTive

Function Sets the active level of the chip select of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:CS:
ACTive {HIGH|LOW}
:TRIGger:ENHanced:SPIBus:CS:ACTive?
```

Example
```
:TRIGGER:ENHANCED:SPIBUS:CS:ACTIVE HIGH
:TRIGGER:ENHANCED:SPIBUS:CS:ACTIVE? ->
:TRIGGER:ENHANCED:SPIBUS:CS:ACTIVE HIGH
```

:TRIGger:ENHanced:SPIBus:CS:SOURce

Function Sets the chip select trace of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:CS:
SOURce {<NRf>}
:TRIGger:ENHanced:SPIBus:CS:SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:SPIBUS:CS:SOURCE 1
:TRIGGER:ENHANCED:SPIBUS:CS:SOURCE? ->
:TRIGGER:ENHANCED:SPIBUS:CS:SOURCE 1
```

:TRIGger:ENHanced:SPIBus:DATA<x>?

Function Queries all settings related to the data of the SPI bus trigger.

Syntax
```
:TRIGger:ENHanced:SPIBus:DATA<x>?
```
<x> = 1 or 2

Example
```
:TRIGGER:ENHANCED:SPIBUS:DATA1? ->
:TRIGGER:ENHANCED:SPIBUS:DATA1:BYTE 1;
CONDITION TRUE;DPOSITION 1;
PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "00010010";SOURCE 3
```

Description DATA2 is valid when `:TRIGger:ENHanced:SPIBus:MODE WIRE4` is specified.

:TRIGger:ENHanced:SPIBus:DATA<x>:BYTE

Function Sets the number of bytes of the data of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:DATA<x>:
BYTE {<NRf>}
:TRIGger:ENHanced:SPIBus:DATA<x>:BYTE?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:SPIBUS:DATA1:BYTE 1
:TRIGGER:ENHANCED:SPIBUS:DATA1:BYTE? ->
:TRIGGER:ENHANCED:SPIBUS:DATA1:BYTE 1
```

:TRIGger:ENHanced:SPIBus:DATA<x>:CONDition

Function Sets the determination method (match or not match) of the data of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:DATA<x>:
CONDition {FALSe|TRUE}
:TRIGger:ENHanced:SPIBus:DATA<x>:
CONDition?
```
<x> = 1 or 2

Example
```
:TRIGGER:ENHANCED:SPIBUS:DATA1:
CONDITION TRUE
:TRIGGER:ENHANCED:SPIBUS:DATA1:
CONDITION? -> :TRIGGER:ENHANCED:SPIBUS:
DATA1:CONDITION TRUE
```

:TRIGger:ENHanced:SPIBus:DATA<x>:DPOSition

Function Sets the pattern comparison start position of the data of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:DATA<x>:
DPOSition {<NRf>}
:TRIGger:ENHanced:SPIBus:DATA<x>:
DPOSition?
```
<x> = 1 or 2
<NRf> = 0 to 9999

Example
```
:TRIGGER:ENHANCED:SPIBUS:DATA1:
DPOSITION 1
:TRIGGER:ENHANCED:SPIBUS:DATA1:
DPOSITION? -> :TRIGGER:ENHANCED:SPIBUS:
DATA1:DPOSITION 1
```

:TRIGger:ENHanced:SPIBus:DATA<x>:HEXA<x>

Function Sets the data of the SPI bus trigger in hexadecimal notation.

Syntax
```
:TRIGger:ENHanced:SPIBus:DATA<x>:HEXA<x> {<String>}
```
<x> of DATA<x> = 1 or 2
<x> of HEXA<x> = 1 to 4
<String> = 2 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:ENHANCED:SPIBUS:DATA1:HEXA1 "AB"
```

:TRIGger:ENHanced:SPIBus:DATA<x>:PATTern<x>

Function Sets the data of the SPI bus trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:DATA<x>:PATTern<x> {<String>}
:TRIGger:ENHanced:SPIBus:DATA<x>:PATTern<x>?
```
<x> of DATA<x> = 1 or 2
<x> of PATTern<x> = 1 to 4
<String> = 8 characters by combining '0,' '1,' and 'X'

Example
```
:TRIGGER:ENHANCED:SPIBUS:DATA1:PATTERN1 "10101011"
:TRIGGER:ENHANCED:SPIBUS:DATA1:PATTERN1? -> :TRIGGER:ENHANCED:SPIBUS:DATA1:PATTERN1 "10101011"
```

:TRIGger:ENHanced:SPIBus:DATA<x>:SOURce

Function Sets the trace of the data of the SPI bus trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:DATA<x>:SOURce {<NRf>}
:TRIGger:ENHanced:SPIBus:DATA<x>:SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:SPIBUS:DATA1:SOURCE 1
:TRIGGER:ENHANCED:SPIBUS:DATA1:SOURCE? -> :TRIGGER:ENHANCED:SPIBUS:DATA1:SOURCE 1
```

:TRIGger:ENHanced:SPIBus:MODE

Function Sets the wiring system of the SPI bus trigger (three-wire or four-wire) or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:MODE {WIRE3|WIRE4}
:TRIGger:ENHanced:SPIBus:MODE?
```

Example
```
:TRIGGER:ENHANCED:SPIBUS:MODE WIRE3
:TRIGGER:ENHANCED:SPIBUS:MODE? -> :TRIGGER:ENHANCED:SPIBUS:MODE WIRE3
```

:TRIGger:ENHanced:TV?

Function Queries all settings related to the TV trigger.

Syntax
```
:TRIGger:ENHanced:TV?
```

Example
```
:TRIGGER:ENHANCED:TV? -> :TRIGGER:ENHANCED:TV:CUSTOMIZE 1;FIELD DONTCARE;FRAME 2;HDTV:LINE 2;POLARITY NEGATIVE;:TRIGGER:ENHANCED:TV:LEVEL 1.000E+00;NTSC:LINE 5;POLARITY NEGATIVE;:TRIGGER:ENHANCED:TV:PAL:LINE 2;POLARITY NEGATIVE;:TRIGGER:ENHANCED:TV:SGUARD 60;SOURCE 1;TYPE HDTV;USERDEFINE:DEFINITION HD;HFREJECTION OFF;HSYNC 50.00E+06;LINE 2;POLARITY NEGATIVE
```

:TRIGger:ENHanced:TV:COUPling?

Function Queries the trigger coupling of the TV trigger.

Syntax
```
:TRIGger:ENHanced:TV:COUPling?
```

Example
```
:TRIGGER:ENHANCED:TV:COUPLING? -> :TRIGGER:ENHANCED:TV:COUPLING TV
```

:TRIGger:ENHanced:TV:CUSTomize

Function Turns ON/OFF the sync guard function of the TV trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:TV:CUSTomize {<Boolean>}
:TRIGger:ENHanced:TV:CUSTomize?
```

Example
```
:TRIGGER:ENHANCED:TV:CUSTOMIZE ON
:TRIGGER:ENHANCED:TV:CUSTOMIZE? -> :TRIGGER:ENHANCED:TV:CUSTOMIZE 1
```

:TRIGger:ENHanced:TV:FIELd

Function Sets the field of the TV trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:TV:FIELd {DONTcare|<NRf>}
:TRIGger:ENHanced:TV:FIELd?
```
<NRf> = 1 or 2

Example
```
:TRIGGER:ENHANCED:TV:FIELD DONTCARE
:TRIGGER:ENHANCED:TV:FIELD? -> :TRIGGER:ENHANCED:TV:FIELD DONTCARE
```

:TRIGger:ENHanced:TV:FRAMe

Function Sets the frame skip function of the TV trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:TV:FRAMe {<NRf>}
:TRIGger:ENHanced:TV:FRAMe?
```
<NRf> = 1, 2, 4, or 8

Example
```
:TRIGGER:ENHANCED:TV:FRAME 2
:TRIGGER:ENHANCED:TV:FRAME? -> :TRIGGER:ENHANCED:TV:FRAME 2
```

:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|USERdefine}?

Function Queries all settings related to the TV trigger mode.

Syntax
```
:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|
USERdefine}?
```

Example (The following is an example for the HDTV.)
```
:TRIGGER:ENHANCED:TV:HDTV? ->
:TRIGGER:ENHANCED:TV:HDTV:LINE 2;
POLARITY NEGATIVE
```

:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL}:HFRejection? (HighFrequencyREJECTION)

Function Queries the low pass filter (HF rejection) of the TV trigger.

Syntax
```
TRIGger:ENHanced:TV:{HDTV|NTSC|
PAL}:HFRejection?
```

Example (The following is an example for the HDTV.)
```
:TRIGGER:ENHANCED:TV:HDTV:HFREJECTION?
-> :TRIGGER:ENHANCED:TV:HDTV:
HFREJECTION OFF
```

:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|USERdefine}:LINE

Function Sets the line for activating the TV trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|
USERdefine}:LINE
:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|
USERdefine}:LINE {<NRf>}
```
<NRf> = 2 to 1251 (for HDTV)
5 to 1054 (for NTSC)
2 to 1251 (for PAL)
2 to 2048 (for USERdefine)

Example (The following is an example for the HDTV.)
```
:TRIGGER:ENHANCED:TV:HDTV:LINE 10
:TRIGGER:ENHANCED:TV:HDTV:LINE? ->
:TRIGGER:ENHANCED:TV:HDTV:LINE 10
```

:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|USERdefine}:POLarity

Function Sets the input polarity of the TV trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|
USERdefine}:POLarity
{NEGative|POSitive}
:TRIGger:ENHanced:TV:{HDTV|NTSC|PAL|
USERdefine}:POLarity?
```

Example (The following is an example for the HDTV.)
```
:TRIGGER:ENHANCED:TV:HDTV:
POLARITY NEGATIVE
:TRIGGER:ENHANCED:TV:HDTV:POLARITY? ->
:TRIGGER:ENHANCED:TV:HDTV:
POLARITY NEGATIVE
```

:TRIGger:ENHanced:TV:LEVel

Function Sets the trigger level of the TV trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:TV:LEVel {<NRf>}
:TRIGger:ENHanced:TV:LEVel?
```
<NRf> = 0.1 to 2.0 (div)

Example
```
:TRIGGER:ENHANCED:TV:LEVEL 1
:TRIGGER:ENHANCED:TV:LEVEL? ->
:TRIGGER:ENHANCED:TV:LEVEL 1.000E+00
```

:TRIGger:ENHanced:TV:SGUard

Function Sets the sync guard of the TV trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:TV:SGUard {<NRf>}
:TRIGger:ENHanced:TV:SGUard?
```
<NRf> = 60 to 90 (%)

Example
```
:TRIGGER:ENHANCED:TV:SGUARD 60
:TRIGGER:ENHANCED:TV:SGUARD? ->
:TRIGGER:ENHANCED:TV:SGUARD 60
```

Description This command is valid when
```
:TRIGGER:ENHANCED:TV:TYPE HDTV|NTSC|
PAL.
```

:TRIGger:ENHanced:TV:SOURce

Function Sets the trigger source of the TV trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:TV:SOURce {<NRf>}
:TRIGger:ENHanced:TV:SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:TV:SOURCE 1
:TRIGGER:ENHANCED:TV:SOURCE? ->
:TRIGGER:ENHANCED:TV:SOURCE 1
```

:TRIGger:ENHanced:TV:TYPE

Function Sets the input type of the TV trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:TV:TYPE {HDTV|NTSC|
PAL|USERdefine}
:TRIGger:ENHanced:TV:TYPE?
```

Example
```
:TRIGGER:ENHANCED:TV:TYPE HDTV
:TRIGGER:ENHANCED:TV:TYPE? ->
TRIGGER:ENHANCED:TV:TYPE HDTV
```

:TRIGger:ENHanced:TV:USERdefine:DEFinition

Function Sets the user-defined resolution or queries the current setting.

Syntax
```
:TRIGger:ENHanced:TV:USERdefine:
DEFinition {HD|SD}
:TRIGger:ENHanced:TV:USERdefine:
DEFinition?
```

Example
```
:TRIGGER:ENHANCED:TV:USERDEFINE:
DEFINITION HD
:TRIGGER:ENHANCED:TV:USERDEFINE:
DEFINITION? -> :TRIGGER:ENHANCED:TV:
USERDEFINE:DEFINITION HD
```

:TRIGger:ENHanced:TV:USERdefine: HFRejection (HighFrequencyREJECTION)

Function Sets the user-defined low pass filter (HF rejection) or queries the current setting.

Syntax
```
:TRIGger:ENHanced:TV:USERdefine:
HFRejection {<Frequency>|OFF}
:TRIGger:ENHanced:TV:USERdefine:
HFRejection?
```
<Frequency> = 300kHz

Example
```
:TRIGGER:ENHANCED:TV:USERDEFINE:
HFREJECTION OFF
:TRIGGER:ENHANCED:TV:USERDEFINE:
HFREJECTION? -> :TRIGGER:ENHANCED:TV:
USERDEFINE:HFREJECTION OFF
```

:TRIGger:ENHanced:TV:USERdefine: HSYNc (Hsync Freq)

Function Sets the user-defined horizontal sync signal or queries the current setting.

Syntax
```
:TRIGger:ENHanced:TV:USERdefine:
HSYNc {<Frequency>}
:TRIGger:ENHanced:TV:USERdefine:HSYNc?
```
<Frequency> = 10k to 200k (Hz)

Example
```
:TRIGGER:ENHANCED:TV:USERDEFINE:
HSYNC 10KHZ
:TRIGGER:ENHANCED:TV:USERDEFINE:HSYNC?
-> :TRIGGER:ENHANCED:TV:USERDEFINE:
HSYNC 10.00E+03
```

:TRIGger:ESTate?

Function Queries all settings related to the edge/state trigger.

Syntax
```
:TRIGger:ESTate?
```

Example
```
:TRIGGER:ESTATE? ->
:TRIGGER:ESTATE:EOR:CHANNEL1 DONTCARE;
CHANNEL2 DONTCARE;CHANNEL3 DONTCARE;
CHANNEL4 DONTCARE;:TRIGGER:ESTATE:
SOURCE 1;POLARITY ENTER
```

:TRIGger:ESTate:EOR?

Function Queries all settings related to the OR trigger.

Syntax
```
:TRIGger:ESTate:EOR?
```

Example
```
:TRIGGER:ESTATE:EOR? ->
:TRIGGER:ESTATE:EOR:CHANNEL1 DONTCARE;
CHANNEL2 DONTCARE;CHANNEL3 DONTCARE;
CHANNEL4 DONTCARE
```

:TRIGger:ESTate:EOR:CHANnel<x>

Function Sets the channel polarity of the OR trigger or queries the current setting.

Syntax
```
:TRIGger:ESTate:EOR:
CHANnel<x> {DONTcare|ENTer|EXIT|FALL|
RISE}
:TRIGger:ESTate:EOR:CHANnel<x>?
```
<x> = 1 to 4

Example
```
:TRIGGER:ESTATE:EOR:CHANNEL1 DONTCARE
:TRIGGER:ESTATE:EOR:CHANNEL1? ->
:TRIGGER:ESTATE:EOR:CHANNEL1 DONTCARE
```

Description
- This command is valid when :TRIGger:TYPE EOR.
- {ENTer|EXIT} is valid when :TRIGger:SOURce:CHANnel<x>:WINDow ON. For all other cases, {FALL|RISE} is valid.

:TRIGger:ESTate:POLarity

Function Sets the polarity of the edge/state trigger or queries the current setting.

Syntax
```
:TRIGger:ESTate:POLarity {ENTer|EXIT|
FALL|RISE}
:TRIGger:ESTate:POLarity?
```

Example
```
:TRIGGER:ESTATE:POLARITY ENTER
:TRIGGER:ESTATE:POLARITY? ->
:TRIGGER:ESTATE:POLARITY ENTER
```

Description
- This command is valid when :TRIGger:TYPE EDGE|EQUalify|STATe.
- This command is invalid when :TRIGger:TYPE EDGE and :TRIGger:ESTate:SOURce LINE.
- {ENTer|EXIT} is valid when :TRIGger:TYPE EDGE|EQUalify and :TRIGger:SOURce:CHANnel<x>:WINDow ON. {FALL|RISE} is valid when :TRIGger:TYPE EDGE|EQUalify and :TRIGger:SOURce:CHANnel<x>:WINDow OFF.
- {ENTer|EXIT} is valid when :TRIGger:TYPE STATe.

:TRIGger:ESTate:SOURce

Function Sets the trigger source of the edge/state trigger or queries the current setting.

Syntax
```
:TRIGger:ESTate:SOURce {<NRf>|EXTernal|
LINE}
:TRIGger:ESTate:SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:ESTATE:SOURCE EXTERNAL
:TRIGGER:ESTATE:SOURCE? ->
:TRIGGER:ESTATE:SOURCE EXTERNAL
```

Description
- This command is valid when :TRIGger:TYPE EDGE|EQUalify.
- {<NRf>|EXTernal|LINE} is valid when :TRIGger:TYPE EDGE.
- {<NRf>|EXTernal} is valid when :TRIGger:TYPE EQUalify.

:TRIGger:HOLDoff

Function Sets the hold off time or queries the current setting.

Syntax
```
:TRIGger:HOLDoff {<Time>}
:TRIGger:HOLDoff?
```
<Time> = 20 ns to 10 s (5 ns steps)

Example
```
:TRIGGER:HOLDOFF 1S
:TRIGGER:HOLDOFF? ->
:TRIGGER:HOLDOFF 1.000E+00
```

:TRIGger:MODE

Function Sets the trigger mode or queries the current setting.

Syntax
```
:TRIGger:MODE {ALEVel|AUTO|NORMal|
NSINgle|SINGle}
:TRIGger:MODE?
```

Example
```
:TRIGGER:MODE ALEVEL
:TRIGGER:MODE? -> :TRIGGER:MODE ALEVEL
```

:TRIGger:POSition

Function Sets the trigger position or queries the current setting.

Syntax
```
:TRIGger:POSition {<NRf>}
:TRIGger:POSition?
```
<NRf> = 0 to 100 (%)

Example
```
:TRIGGER:POSITION 10
:TRIGGER:POSITION? ->
:TRIGGER:POSITION 10
```

: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> = See the DL9000 User's Manual.

Example
```
:TRIGGER:SCOUNT 1
:TRIGGER:SCOUNT? -> :TRIGGER:SCOUNT 1
```

:TRIGger:SOURce?

Function Queries all settings related to the trigger source.

Syntax
```
:TRIGger:SOURce?
```

Example
```
:TRIGGER:SOURCE? -> :TRIGGER:SOURCE:
CHANNEL1:COUPLING DC;HFREJECTION OFF;
HYSTERESIS HIGH;LEVEL 1.000E+00;
STATE HIGH;WIDTH 1.000E+00;WINDOW 0;:
TRIGGER:SOURCE:CHANNEL2:COUPLING DC;
HFREJECTION OFF;HYSTERESIS HIGH;
LEVEL 1.000E+00;STATE HIGH;
WIDTH 1.000E+00;WINDOW 0;:TRIGGER:
SOURCE:CHANNEL3:COUPLING DC;
HFREJECTION OFF;HYSTERESIS HIGH;
LEVEL 1.000E+00;STATE HIGH;
WIDTH 1.000E+00;WINDOW 0;:TRIGGER:
SOURCE:CHANNEL4:COUPLING DC;
HFREJECTION OFF;HYSTERESIS HIGH;
LEVEL 1.000E+00;STATE HIGH;
WIDTH 1.000E+00;WINDOW 0;:TRIGGER:
SOURCE:EXTERNAL:LEVEL 0.000E+00;
PROBE 1;:TRIGGER:SOURCE:LOGIC AND
```

:TRIGger:SOURce:CHANnel<x>?

Function Queries all settings related to the channel of the trigger source.

Syntax
```
:TRIGger:SOURce:CHANnel<x>?
```
<x> = 1 to 4

Example
```
:TRIGGER:SOURCE:CHANNEL1? ->
:TRIGGER:SOURCE:CHANNEL1:COUPLING DC;
HFREJECTION OFF;HYSTERESIS HIGH;
LEVEL 1.000E+00;STATE HIGH;
WIDTH 1.000E+00;WINDOW 0
```

:TRIGger:SOURce:CHANnel<x>:COUPling

Function Sets the trigger coupling of the channel or queries the current setting.

Syntax
```
:TRIGger:SOURce:CHANnel<x>:
COUPling {AC|DC}
:TRIGger:SOURce:CHANnel<x>:COUPling?
```
<x> = 1 to 4

Example
```
:TRIGGER:SOURCE:CHANNEL1:COUPLING AC
:TRIGGER:SOURCE:CHANNEL1:COUPLING? ->
:TRIGGER:SOURCE:CHANNEL1:COUPLING DC
```

:TRIGger:SOURce:CHANnel<x>: HFRejection (HighFrequencyREJECTION)

Function Sets the low pass filter (HF rejection) of the channel or queries the current setting.

Syntax
```
:TRIGger:SOURce:CHANnel<x>:
HFRejection {<Frequency>|OFF}
:TRIGger:SOURce:CHANnel<x>:HFRejection?
```
<x> = 1 to 4
<Frequency> = 20MHz or 15kHz

Example
```
:TRIGGER:SOURCE:CHANNEL1:
HFREJECTION OFF
:TRIGGER:SOURCE:CHANNEL1:HFREJECTION?
-> :TRIGGER:SOURCE:CHANNEL1:
HFREJECTION OFF
```

Description This command is invalid when the trigger source is {EXTernal|LINE}.

:TRIGger:SOURce:CHANnel<x>: HYSTeresis

Function Sets the hysteresis of the channel or queries the current setting.

Syntax
```
:TRIGger:SOURce:CHANnel<x>:
HYSTeresis {HIGH|LOW}
:TRIGger:SOURce:CHANnel<x>:HYSTeresis?
```
<x> = 1 to 4

Example
```
: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 channel or queries the current setting.

Syntax
```
:TRIGger:SOURce:CHANnel<x>:
LEVel {<Voltage>|<Current>}
:TRIGger:SOURce:CHANnel<x>:LEVel?
```
<x> = 1 to 4
<Voltage> and <Current> = See the DL9000 User's Manual.

Example
```
:TRIGGER:SOURCE:CHANNEL1:LEVEL 1V
:TRIGGER:SOURCE:CHANNEL1:LEVEL? ->
:TRIGGER:SOURCE:CHANNEL1:
LEVEL 1.000E+00
```

:TRIGger:SOURce:CHANnel<x>:STATe

Function Sets the condition to be satisfied of the channel or queries the current setting.

Syntax
```
:TRIGger:SOURce:CHANnel<x>:
STATe {DONTcare|HIGH|IN|LOW|OUT}
:TRIGger:SOURce:CHANnel<x>:STATe?
```
<x> = 1 to 4

Example
```
:TRIGGER:SOURCE:CHANNEL1:STATE HIGH
:TRIGGER:SOURCE:CHANNEL1:STATE? ->
:TRIGGER:SOURCE:CHANNEL1:STATE HIGH
```

Description
- This command is valid when :TRIGger:TYPE EQUalify|I2CBus|PQUalify|PSTAte|SPATtern|STATe.
- {HIGH|LOW} is valid when :TRIGger:TYPE I2CBus|SPATtern.
- {IN|OUT} is valid when :TRIGger:TYPE EQUalify|PQUalify|PSTAte|STATe and :TRIGger:SOURce:CHANnel<x>:WINDow ON.
  {HIGH|LOW} is valid when :TRIGger:TYPE EQUalify|PQUalify|PSTAte|STATe and :TRIGger:SOURce:CHANnel<x>:WINDow OFF.

:TRIGger:SOURce:CHANnel<x>:WIDTh

Function Sets the window trigger width of the channel or queries the current setting.

Syntax
```
:TRIGger:SOURce:CHANnel<x>:
WIDTh {<Voltage>|<Current>}
:TRIGger:SOURce:CHANnel<x>:WIDTh?
```
<x> = 1 to 4
<Voltage> and <Current> = See the DL9000 User's Manual.

Example
```
:TRIGGER:SOURCE:CHANNEL1:WIDTH 1V
:TRIGGER:SOURCE:CHANNEL1:WIDTH? ->
:TRIGGER:SOURCE:CHANNEL1:WIDTH
1.000E+00
```

Description This command is valid when :TRIGger:SOURce:CHANnel<x>:WINDow ON.

:TRIGger:SOURce:CHANnel<x>:WINDow

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

Syntax
```
:TRIGger:SOURce:CHANnel<x>:
WINDow {<Boolean>}
:TRIGger:SOURce:CHANnel<x>:WINDow?
```
<x> = 1 to 4

Example
```
:TRIGGER:SOURCE:CHANNEL1:WINDOW ON
:TRIGGER:SOURCE:CHANNEL1:WINDOW? ->
:TRIGGER:SOURCE:CHANNEL1:WINDOW 1
```

:TRIGger:SOURce:EXTernal?

Function Queries all settings related to the external trigger.

Syntax
```
:TRIGger:SOURce:EXTernal?
```

Example
```
:TRIGGER:SOURCE:EXTERNAL? ->
:TRIGGER:SOURCE:EXTERNAL:
LEVEL 0.000E+00;PROBE 1
```

:TRIGger:SOURce:EXTernal:LEVel

Function Sets the trigger level of the external trigger or queries the current setting.

Syntax
```
:TRIGger:SOURce:EXTernal:
LEVel {<Voltage>|<Current>}
:TRIGger:SOURce:EXTernal:LEVel?
```
<x> = 1 to 4
<Voltage> and <Current> = See the DL9000 User's Manual.

Example
```
:TRIGGER:SOURCE:EXTERNAL:LEVEL 1V
:TRIGGER:SOURCE:EXTERNAL:LEVEL? ->
:TRIGGER:SOURCE:EXTERNAL:
LEVEL 1.000E+00
```

Description This command is valid when `:TRIGger:TYPE EDGE|EQUalify|PQUalify|PULSe`.

:TRIGger:SOURce:EXTernal:PROBe

Function Sets the probe attenuation of the external trigger or queries the current setting.

Syntax
```
:TRIGger:SOURce:EXTernal:PROBe {<NRf>}
:TRIGger:SOURce:EXTernal:PROBe?
```
<NRf> = 1,10

Example
```
:TRIGGER:SOURCE:EXTERNAL:PROBE 1
:TRIGGER:SOURCE:EXTERNAL:PROBE? ->
:TRIGGER:SOURCE:EXTERNAL:PROBE 1
```

Description This command is valid when `:TRIGger:TYPE EDGE|EQUalify|PQUalify|PULSe`.

:TRIGger:SOURce:LOGic

Function Sets the trigger source logic or queries the current setting.

Syntax
```
:TRIGger:SOURce:LOGic {AND|OR}
:TRIGger:SOURce:LOGic?
```

Example
```
:TRIGGER:SOURCE:LOGIC AND
:TRIGGER:SOURCE:LOGIC? ->
:TRIGGER:SOURCE:LOGIC AND
```

Description This command is valid when `:TRIGger:TYPE EQUalify|I2CBus|PQUalify|PSTAte|SPATtern|STATe`.

:TRIGger:TYPE

Function Sets the trigger type or queries the current setting.

Syntax
```
:TRIGger:TYPE {EDGE|EICYcle|EIDelay|
EISequence|EOR|EQUalify|I2CBus|
PQUalify|PSTAte|PULSe|SPATtern|SPIBus|
STATe|TV}
:TRIGger:TYPE?
```

Example
```
:TRIGGER:TYPE EDGE
:TRIGGER:TYPE? -> :TRIGGER:TYPE EDGE
```

:TRIGger:WIDTh?

Function Queries all settings related to the pulse width trigger.

Syntax
```
:TRIGger:WIDTh?
```

Example
```
:TRIGGER:WIDTH? -> :TRIGGER:WIDTH:
MODE OUT;POLARITY POSITIVE;SOURCE 1;
TIME1 1.000E-09;TIME2 1.000E-09
```

:TRIGger:WIDTh:MODE

Function Sets the determination mode of the pulse width trigger or queries the current setting.

Syntax
```
:TRIGger:WIDTh:MODE {BETWeen|IN|
NOTBetween|OUT|TIMeout}
:TRIGger:WIDTh:MODE?
```

Example
```
:TRIGGER:WIDTH:MODE BETWEEN
:TRIGGER:WIDTH:MODE? ->
:TRIGGER:WIDTH:MODE BETWEEN
```

:TRIGger:WIDTh:POLarity

Function Sets the polarity of the pulse width trigger or queries the current setting.

Syntax
```
:TRIGger:WIDTh:POLarity {FALSe|IN|
NEGative|OUT|POSitive|TRUE}
:TRIGger:WIDTh:POLarity?
```

Example
```
:TRIGGER:WIDTH:POLARITY POSITIVE
:TRIGGER:WIDTH:POLARITY? ->
:TRIGGER:WIDTH:POLARITY POSITIVE
```

Description
- `{IN|OUT}` is valid when `:TRIGger:TYPE PQUalify|PULSe` and `:TRIGger:SOURce:CHANnel<x>:WINDow ON`.
- `{HIGH|LOW}` is valid when `:TRIGger:TYPE PQUalify|PULSe` and `:TRIGger:SOURce:CHANnel<x>:WINDow OFF`.
- `{FALSe|TRUE}` is valid when `:TRIGger:TYPE PSTATe`.

:TRIGger:WIDTh:SOURce

Function Sets the trigger source of the pulse width trigger or queries the current setting.

Syntax
```
:TRIGger:WIDTh:SOURce {<NRf>|EXTernal}
:TRIGger:WIDTh:SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:WIDTH:SOURCE EXTERNAL
:TRIGGER:WIDTH:SOURCE? ->
:TRIGGER:WIDTH:SOURCE EXTERNAL
```

Description This command is valid when `:TRIGger:TYPE PQUalify|PULSe`.

:TRIGger:WIDTh:TIME<x>

Function Sets the pulse width of the pulse width trigger or queries the current setting.

Syntax
```
:TRIGger:WIDTh:TIME<x> {<Time>}
:TRIGger:WIDTh:TIME<x>?
```
<x> = 1 or 2
<Time> = 1 ns to 10 s (500 ps steps)

Example
```
:TRIGGER:WIDTH:TIME1 1S
:TRIGGER:WIDTH:TIME1? ->
:TRIGGER:WIDTH:TIME1 1.000E+00
```

Description TIME2 is valid when `:TRIGger:WIDTh:MODE BETWeen|NOTBetween`.

5.30 WAVeform Group

The commands in this group deal with 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 1;
RECORD 0;START 0;END 6249999;
FORMAT WORD;
BYTEORDER LSBFIRST

: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:END

Function Sets the last data point of the waveform specified by :WAVeform:TRACe or queries the current setting.
Syntax :WAVeform:END {<NRf>}
:WAVeform:END?
<NRf> = 0 to 6,249,999
Example :WAVEFORM:END 12499
:WAVEFORM:END? -> :WAVEFORM:END 12499
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 ASCII
:WAVEFORM:FORMAT? ->
:WAVEFORM:FORMAT ASCII
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 waveform specified by ":WAVeform:TRACe".
Syntax :WAVeform:LENGth?
Example :WAVEFORM:LENGTH? ->
:WAVEFORM:LENGTH 12500

:WAVeform:OFFSet?

Function Queries the offset value when converting the waveform data specified by :WAVeform:TRACe to physical values.
Syntax :WAVeform:OFFSet?
Example :WAVEFORM:OFFSET? -> 0.000E+00
Description
- The offset value is used when converting the <Block data> that is output using :WAVeform:SEND? to physical values.
- When :CHANnel<x>:OCANcel is ON, 0 is returned.

: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? -> 5.000E+00
Description The range 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 for the commands in the WAVeform group or queries the current setting.
Syntax :WAVeform:RECord {AVERage|MINimum|<NRf>}
:WAVeform:RECord?
<NRf> = 0 to –1999
Example :WAVEFORM:RECORD 0
:WAVEFORM:RECORD? -> :WAVEFORM:RECORD 0
Description
- If "AVERage" is specified, the commands in the WAVeform group are applied to the average value of the history waveform. The record numbers to be averaged are set using the ":HISTory[:CURRent]:DISPlay" command. In addition, the highlight display mode must be set to "AVERage." Set the highlight display mode using the ":HISTory[:CURent]:MODE" command.
- Specifying "MINimum" sets the record to the minimum record number. The selectable record number varies depending on the model and acquisition setting. For details, see the DL9000 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 –1999

:WAVeform:SEND?

Function Queries the waveform data specified by ":WAVeform:TRACe".

Syntax :WAVeform:SEND? [{<NRf>}]
<NRf> = 1 to 2000
Varies depending on the record length setting.

Example :WAVEFORM:SEND? -> #8 (number of bytes, 8 digits) (data sequence)
or <NRf>,<NRf>,...

Description
- The output format of :WAVeform:SEND? varies depending on the :WAVeform:FORMat setting.
  (1) When set to ASCii
  Returned in the following format: <Voltage>, <Voltage>, ... <Voltage>.
  (2) When set to BYTE or WORD
  Returned in the <Block data> format.
  You can convert the value using the following equation.
  Voltage (computed value) = (range × data / divisions*) + offset
  * BYTE: Divisions = 12.5
  WORD: Divisions = 3200
- <NRf> can be omitted. If <NRf> is attached, waveform data is queried <NRf> times in order from the record number specified by :WAVeform:RECord – <NRf> + 1.

:WAVeform:SIGN?

Function Queries the existence of a sign when querying the waveform data specified by :WAVeform:TRACe using binary data.

Syntax :WAVeform:SIGN?

Example :WAVEFORM:SIGN? -> :WAVEFORM:SIGN 1

:WAVeform:SRATe? (Sample RATE)

Function Queries the sample rate of the record specified by :WAVeform:RECord.

Syntax :WAVeform:SRATe?

Example :WAVEFORM:SRATE? ->
:WAVEFORM:SRATE 1.25E+09

:WAVeform:STARt

Function Sets the first data point of the waveform specified by :WAVeform:TRACe or queries the current setting.

Syntax :WAVeform:STARt {<NRf>}
:WAVeform:STARt?
<NRf> = 0 to 6,249,999 (0 to 2,499,999 on 2.5 MW memory models)

Example :WAVEFORM:START 0
:WAVEFORM:START? -> :WAVEFORM:START 0

:WAVeform:TRACe

Function Sets the target waveform or queries the current setting.

Syntax :WAVeform:TRACe {<NRf>|MATH<x>|REFerence<x>}
:WAVeform:TRACe?
<NRf> = 1 to 4
<x> of MATH<x> = 1 to 8
<x> of REFerence<x> = 1 to 4

Example :WAVEFORM:TRACE 1
:WAVEFORM:TRACE? -> :WAVEFORM:TRACE 1

:WAVeform:TRIGger?

Function Queries the trigger position of the record specified by :WAVeform:RECord.

Syntax :WAVeform:TRIGger?

Example :WAVEFORM:TRIGGER? ->
:WAVEFORM:TRIGGER 6250

Description Queries the number of points from the first point of the record length to the trigger position.

:WAVeform:TYPE?

Function Queries the acquisition mode of the waveform specified by :WAVeform:TRACe.

Syntax :WAVeform:TYPE?

Example :WAVEFORM:TYPE? ->
:WAVEFORM:TYPE NORMAL

5.31 ZOOM Group

:ZOOM?

Function Queries all settings related to the waveform zoom.

Syntax :ZOOM?

Example :ZOOM? -> :ZOOM:ALLOCATION1:TRACE1 1;
TRACE2 1;TRACE3 1;TRACE4 1;TRACE5 1;
TRACE6 1;TRACE7 1;TRACE8 1;:ZOOM:
ALLOCATION2:TRACE1 1;TRACE2 1;TRACE3 1;
TRACE4 1;TRACE5 1;TRACE6 1;TRACE7 1;
TRACE8 1;:ZOOM:FORMAT1 MAIN;
FORMAT2 MAIN;HLINKAGE 0;HORIZONTAL1:
ASCROLL:SPEED 5;:ZOOM:HORIZONTAL1:
MAG 2.000E+00;POSITION 0.000E+00;:ZOOM:
HORIZONTAL2:ASCROLL:SPEED 5;:ZOOM:
HORIZONTAL2:MAG 2.000E+00;
POSITION 0.000E+00;:ZOOM:MODE MAIN;
TYPE1 HORIZONTAL;TYPE2 HORIZONTAL;
VERTICAL1:MAG 1.000E+00;
POSITION 0.000E+00;TRACE 1;:ZOOM:
VERTICAL2:MAG 1.000E+00;
POSITION 0.000E+00;TRACE 1;:ZOOM:
VLINKAGE 0

:ZOOM:ALLocation<x>?

Function Queries all settings related to the zoom source waveform.

Syntax :ZOOM:ALLocation<x>?
<x> = 1 or 2

Example :ZOOM:ALLOCATION1? ->
:ZOOM:ALLOCATION1:TRACE1 1;TRACE2 1;
TRACE3 1;TRACE4 1;TRACE5 1;TRACE6 1;
TRACE7 1;TRACE8 1

:ZOOM:ALLocation<x>:ALLon

Function Sets all waveforms to be zoomed.

Syntax :ZOOM:ALLocation<x>:ALLon
<x> = 1 or 2

Example :ZOOM:ALLOCATION1:ALLON

:ZOOM:ALLocation<x>:TRACe<x>

Function Turns ON/OFF the trace you wish to zoom or queries the current setting.

Syntax :ZOOM:ALLocation<x>:
TRACe<x> {<Boolean>}
:ZOOM:ALLocation<x>:TRACe<x>?
<x> of ALLocation<x> = 1 or 2
<x> of TRACe<x> = 1 to 8

Example :ZOOM:ALLOCATION1:TRACE1 ON
:ZOOM:ALLOCATION1:TRACE1? ->
:ZOOM:ALLOCATION1:TRACE1 1

:ZOOM:FORMat<x>

Function Sets the display format of the zoom waveform or queries the current setting.

Syntax :ZOOM:FORMat<x> {DUAL|MAIN|QUAD|SINGle|
TRIad}
:ZOOM:FORMat<x>?
<x> = 1 or 2

Example :ZOOM:FORMAT1 SINGLE
:ZOOM:FORMAT1? -> :ZOOM:FORMAT1 SINGLE

:ZOOM:HLINkage

Function Turns ON/OFF the horizontal link or queries the current setting.

Syntax :ZOOM:HLINkage {<Boolean>}
:ZOOM:HLINkage?

Example :ZOOM:HLINKAGE ON
:ZOOM:HLINKAGE? -> :ZOOM:HLINKAGE 1

:ZOOM:HORizontal<x>?

Function Queries all settings related to the horizontal zoom.

Syntax :ZOOM:HORizontal<x>?
<x> = 1 or 2

Example :ZOOM:HORIZONTAL1? ->
:ZOOM:HORIZONTAL1:ASCROLL:SPEED 5;:
ZOOM:HORIZONTAL1:MAG 2.000E+00;
POSITION 4.000E+00

:ZOOM:HORizontal<x>:ASCRoll?

Function Queries all settings related to the auto scroll function.

Syntax :ZOOM:HORizontal<x>:ASCRoll?
<x> = 1 or 2

Example :ZOOM:HORIZONTAL1:ASCROLL? ->
:ZOOM:HORIZONTAL1:ASCROLL:SPEED 5

:ZOOM:HORizontal<x>:ASCRoll:JUMP

Function Moves the zoom center position to the left or right edge of the main screen.

Syntax :ZOOM:HORizontal<x>:ASCRoll:JUMP {LEFT|
RIGHt}
<x> = 1 or 2

Example :ZOOM:HORIZONTAL1:ASCROLL:JUMP RIGHT

:ZOOM:HORizontal<x>:ASCRoll:SPEed

Function Sets the auto scroll speed or queries the current setting.

Syntax :ZOOM:HORizontal<x>:ASCRoll:SPEed
{<NRf>}
:ZOOM:HORizontal<x>:ASCRoll:SPEed?
<x> = 1 or 2
<NRf> = 1, 2, 5, 10, 20, 50

Example :ZOOM:HORIZONTAL1:ASCROLL:SPEED 1
:ZOOM:HORIZONTAL1:ASCROLL:SPEED? ->
:ZOOM:HORIZONTAL1:ASCROLL:SPEED 1

:ZOOM:HORizontal<x>:ASCRoll:STARt

Function Starts auto scrolling.

Syntax
```
:ZOOM:HORizontal<x>:ASCRoll:STARt
{LEFT|RIGHt}
```
<x> = 1 or 2

Example
```
:ZOOM:HORIZONTAL1:ASCROLL:START LEFT
```

:ZOOM:HORizontal<x>:ASCRoll:STOP

Function Stops auto scrolling.

Syntax
```
:ZOOM:HORizontal<x>:ASCRoll:STOP
```
<x> = 1 or 2

Example
```
:ZOOM:HORIZONTAL1:ASCROLL:STOP
```

:ZOOM:HORizontal<x>:MAG

Function Sets the horizontal zoom magnification or queries the current setting.

Syntax
```
:ZOOM:HORizontal<x>:MAG {<NRf>}
:ZOOM:HORizontal<x>:MAG?
```
<x> = 1 or 2

<NRf> = See the DL9000 User's Manual.

Example
```
:ZOOM:HORIZONTAL1:MAG 2
:ZOOM:HORIZONTAL1:MAG? ->
:ZOOM:HORIZONTAL1:MAG 2.000E+00
```

:ZOOM:HORizontal<x>:POSition

Function Sets the horizontal zoom center position or queries the current setting.

Syntax
```
:ZOOM:HORizontal<x>:POSition {<NRf>}
:ZOOM:HORizontal<x>:POSition?
```
<x> = 1 or 2

<NRf> = –5 to 5 (div)

Example
```
:ZOOM:HORIZONTAL1:POSITION 1
:ZOOM:HORIZONTAL1:POSITION? ->
:ZOOM:HORIZONTAL1:POSITION 1.000E+00
```

:ZOOM:MODE

Function Sets the zoom waveform display format or queries the current setting.

Syntax
```
:ZOOM:MODE {MAIN|MAIN_Z1|MAIN_Z1_Z2|
MAIN_Z2|Z1|Z1_Z2|Z2}
:ZOOM:MODE?
```

Example
```
:ZOOM:MODE MAIN_Z1_Z2
:ZOOM:MODE? -> :ZOOM:MODE MAIN_Z1_Z2
```

:ZOOM:TYPE<x>

Function Sets the zoom type or queries the current setting.

Syntax
```
:ZOOM:TYPE<x> {HORizontal|VERTical}
:ZOOM:TYPE<x>?
```
<x> = 1 or 2

Example
```
:ZOOM:TYPE1 VERTICAL
:ZOOM:TYPE1? -> :ZOOM:TYPE1 VERTICAL
```

:ZOOM:VERTical<x>?

Function Queries all settings related to the vertical zoom.

Syntax
```
:ZOOM:VERTical<x>?
```
<x> = 1 or 2

Example
```
:ZOOM:VERTICAL1? -> :ZOOM:VERTICAL1:
MAG 1.000E+00;POSITION 0.000E+00;
TRACE 1
```

:ZOOM:VERTical<x>:INITialize

Function Initializes the vertical zoom.

Syntax
```
:ZOOM:VERTical<x>:INITialize
```
<x> = 1 or 2

Example
```
:ZOOM:VERTICAL1:INITIALIZE
```

:ZOOM:VERTical<x>:MAG

Function Sets the vertical zoom magnification or queries the current setting.

Syntax
```
:ZOOM:VERTical<x>:MAG {<NRf>}
:ZOOM:VERTical<x>:MAG?
```
<x> = 1 or 2

<NRf> = See the DL9000 User's Manual.

Example
```
:ZOOM:VERTICAL1:MAG 1
:ZOOM:VERTICAL1:MAG? ->
:ZOOM:VERTICAL1:MAG 1.000E+00
```

:ZOOM:VERTical<x>:POSition

Function Sets the vertical zoom position or queries the current setting.

Syntax
```
:ZOOM:VERTical<x>:POSition {<NRf>}
:ZOOM:VERTical<x>:POSition?
```
<x> = 1 or 2

<NRf> = –4 to 4 (div)

Example
```
:ZOOM:VERTICAL1:POSITION 1
:ZOOM:VERTICAL1:POSITION? ->
:ZOOM:VERTICAL1:POSITION 1.000E+00
```

:ZOOM:VERTical<x>:TRACe

Function Sets the trace you wish to display on the vertical zoom screen or queries the current setting.

Syntax
```
:ZOOM:VERTical<x>:TRACe {<NRf>}
:ZOOM:VERTical<x>:TRACe?
```
<x> = 1 or 2

<NRf> = 1 to 8

Example
```
:ZOOM:VERTICAL1:TRACE 1
:ZOOM:VERTICAL1:TRACE? ->
:ZOOM:VERTICAL1:TRACE 1
```

:ZOOM:VLINkage

Function Turns ON/OFF the vertical link or queries the current setting.

Syntax
```
:ZOOM:VLINkage {<Boolean>}
:ZOOM:VLINkage?
```

Example
```
:ZOOM:VLINkAGE ON
:ZOOM:VLINkAGE? -> :ZOOM:VLINkAGE 1
```

5.32 Common Command Group

The commands in the common group are defined in the USBTMC-USB488 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 6.

*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 6-3.

*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 6-3.

*IDN? (IDeNtify)

Description Queries the instrument model.
Syntax *IDN?
Example *IDN? ->
YOKOGAWA,701313,27E100000,F1.10
Description The information is returned in the following form: <Manufacturer>,<Model>,<Serial No.>,<Firmware version> The values 701310, 701311, 701312, and 701313 are returned for the <Model> when the instrument is the DL9140, DL9140L, DL9240, and DL9240L, respectively.

***LRN? (LeaRN)**

Function Queries collectively the current settings of the following command groups.
ACQuire, CHANnel<x>, TIMebase, TRIGger

Syntax *LRN?

Example *LRN? -> :ACQUIRE:AVERAGE:COUNT 2;
EWEIGHT 16;:ACQUIRE:HRMODE 0;
INTERLEAVE 0;INTERPOLATE 1;MODE NORMAL;
REPETITIVE 0;RLENGTH 12500;:CHANNEL1:
SELECT INPUT;DISPLAY 1;BWIDTH FULL;
COUPLING DC;DESKEW 0.000E+00;INVERT 0;
LABEL:DEFINE "CH1";MODE 1;:CHANNEL1:
OCANCEL 0;OFFSET 0.000E+00;
POSITION 0.000E+00;PROBE:MODE 1;:
CHANNEL1:SVALUE 0;VDIV 1.000E+00;:
CHANNEL2:SELECT INPUT;DISPLAY 1;
BWIDTH FULL;COUPLING DC;
DESKEW 0.000E+00;INVERT 0;LABEL:
DEFINE "CH2";MODE 1;:CHANNEL2:
OCANCEL 0;OFFSET 0.000E+00;
POSITION 0.000E+00;PROBE:MODE 1;:
CHANNEL2:SVALUE 0;VDIV 1.000E+00;:
CHANNEL3:SELECT INPUT;DISPLAY 1;
BWIDTH FULL;COUPLING DC;
DESKEW 0.000E+00;INVERT 0;LABEL:
DEFINE "CH3";MODE 1;:CHANNEL3:
OCANCEL 0;OFFSET 0.000E+00;
POSITION 0.000E+00;PROBE:MODE 1;:
CHANNEL3:SVALUE 0;VDIV 1.000E+00;:
CHANNEL4:SELECT INPUT;DISPLAY 1;
BWIDTH FULL;COUPLING DC;
DESKEW 0.000E+00;INVERT 0;LABEL:
DEFINE "CH4";MODE 1;:CHANNEL4:
OCANCEL 0;OFFSET 0.000E+00;
POSITION 0.000E+00;PROBE:MODE 1;:
CHANNEL4:SVALUE 0;VDIV 1.000E+00;:
TIMEBASE:TDIV 1.000E-06;:TRIGGER:
ACTION:ACQCOUNT 1;BUZZER 0;HCOPY 0;
MODE OFF;SAVE 0;:TRIGGER:TYPE EDGE;
CLOCK:SOURCE 1;POLARITY RISE;:TRIGGER:
DELAY:EDGECOUNT:COUNT 1;:TRIGGER:DELAY:
MODE 0;POLARITY RISE;SOURCE 1;
TIME 0.000E+00;TYPE BYTIME;:TRIGGER:
EINTERVAL:EVENT1:TYPE EDGE;CLOCK:
SOURCE 1;POLARITY RISE;:TRIGGER:
EINTERVAL:EVENT1:ESTATE:SOURCE 1;
POLARITY RISE;:TRIGGER:EINTERVAL:
EVENT1:STATE:CHANNEL1 DONTCARE;
CHANNEL2 DONTCARE;CHANNEL3 DONTCARE;
CHANNEL4 DONTCARE;LOGIC AND;:TRIGGER:
EINTERVAL:EVENT1:WIDTH:MODE OUT;
POLARITY POSITIVE;SOURCE 1;
TIME1 1.000E-09;TIME2 2.000E-09;:
TRIGGER:EINTERVAL:EVENT2:TYPE EDGE;
CLOCK:SOURCE 1;POLARITY RISE;:TRIGGER:
EINTERVAL:EVENT2:ESTATE:SOURCE 1;
POLARITY RISE;:TRIGGER:EINTERVAL:
EVENT2:STATE:CHANNEL1 DONTCARE;
CHANNEL2 DONTCARE;CHANNEL3 DONTCARE;
CHANNEL4 DONTCARE;LOGIC AND;:TRIGGER:
EINTERVAL:EVENT2:WIDTH:MODE OUT;
POLARITY POSITIVE;SOURCE 1;
TIME1 1.000E-09;TIME2 2.000E-09;:
TRIGGER:EINTERVAL:MODE OUT;
TIME1 1.500E-09;TIME2 2.000E-09;TRY:
MODE 0;SELECT 1;:TRIGGER:ENHANCED:TV:
CUSTOMIZE 0;FIELD 1;FRAME 1;HDTV:
LINE 2;POLARITY POSITIVE;:TRIGGER:
ENHANCED:TV:LEVEL 500.0E-03;NTSC:
LINE 5;POLARITY NEGATIVE;:TRIGGER:
ENHANCED:TV:PAL:LINE 2;
POLARITY NEGATIVE;:TRIGGER:ENHANCED:TV:
SGUARD 75;SOURCE 1;TYPE NTSC;
USERDEFINE:DEFINITION HD;
HFREJECTION OFF;HSYNC 31.500E+03;
LINE 2;POLARITY POSITIVE;:TRIGGER:
ESTATE:EOR:CHANNEL1 RISE;CHANNEL2 RISE;
CHANNEL3 RISE;CHANNEL4 RISE;:TRIGGER:
ESTATE:SOURCE 1;POLARITY RISE;:TRIGGER:
HOLDOFF 20.00E-09;MODE AUTO;
POSITION 50;SCOUNT 1;SOURCE:CHANNEL1:
COUPLING DC;HFREJECTION OFF;
HYSTERESIS LOW;LEVEL 0.000E+00;
STATE DONTCARE;WIDTH 1.000E+00;
WINDOW 0;:TRIGGER:SOURCE:CHANNEL2:
COUPLING DC;HFREJECTION OFF;
HYSTERESIS LOW;LEVEL 0.000E+00;
STATE DONTCARE;WIDTH 1.000E+00;
WINDOW 0;:TRIGGER:SOURCE:CHANNEL3:
COUPLING DC;HFREJECTION OFF;
HYSTERESIS LOW;LEVEL 0.000E+00;
STATE DONTCARE;WIDTH 1.000E+00;
WINDOW 0;:TRIGGER:SOURCE:CHANNEL4:
COUPLING DC;HFREJECTION OFF;
HYSTERESIS LOW;LEVEL 0.000E+00;
STATE DONTCARE;WIDTH 1.000E+00;
WINDOW 0;:TRIGGER:SOURCE:EXTERNAL:
LEVEL 0.000E+00;PROBE 1;:TRIGGER:
SOURCE:LOGIC AND;:TRIGGER:WIDTH:
MODE OUT;POLARITY POSITIVE;SOURCE 1;
TIME1 1.000E-09;TIME2 2.000E-09

*OPC (OPeration Complete)

Function Sets bit 0 (OPC bit) of the standard event register to 1 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 5-7.
- 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.

*OPT? (OPTion)

Description Queries the installed options.

Syntax `*OPT?`

Example `*OPT? -> CH6.25MW,PRINTER,ETHER,HDD, USERDEFINE,I2C,PROBEPOWER,SCSI,ETHER, USERDEFINE`

Description
- Returns the memory model and the presence/absence of the built-in printer, Ethernet, internal hard disk, user-defined computation, I^2C analysis function, and rear panel probe power.
- 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 6.

*RST (ReSeT)

Function Initializes the settings.

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? -> 239`

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 6-1.

*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 5-2.

*TST?

Function Performs a self-test and queries the result. The self test involves internal memory tests.

Syntax `*TST?`

Example `*TST? -> 0`

Description If the self-test is successful, 0 is returned. If there is an error, 1 is returned.

*WAI (WAIt)

Function Holds the subsequent command until the completion of the specified overlap operation.

Syntax `*WAI`

Example `*WAI`

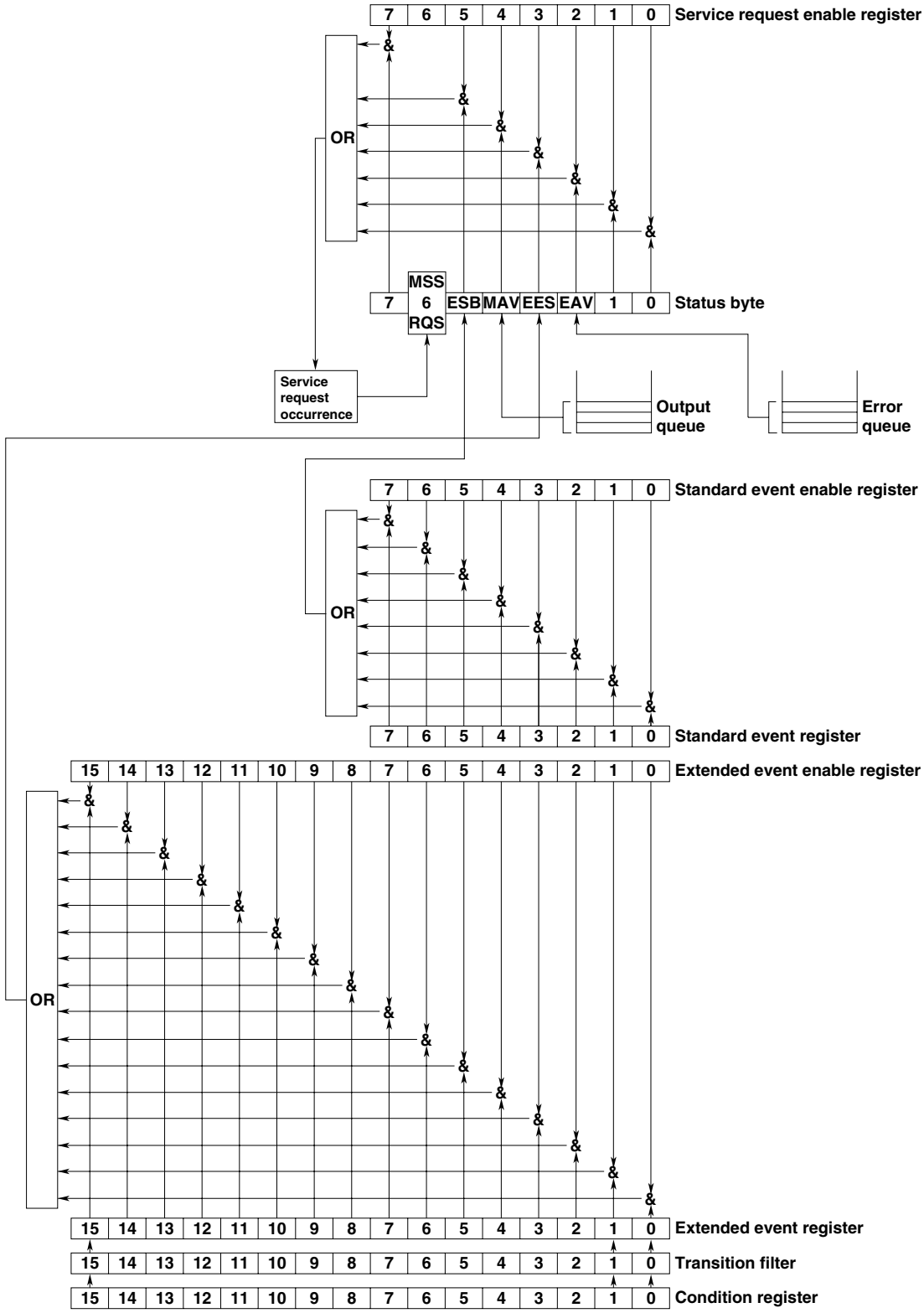
Description
- For the description regarding how to synchronize the program using *WAI, see page 5-7.
- The `:COMMunicate:OPSE` command is used to specify the overlap command.

Chapter 6 Status Reports

6.1 Overview of the Status Report

Status Reports

The figure below shows the status report that is read by serial polling. This status report is an extended version of the status report defined in IEEE 488.2-1992.



Overview of the Registers and Queues

Name	Functions	Writing	Reading
Status byte		–	Serial polling (RQS), *STB?(MSS)
Service request enable register	Masks status byte	*SRE	*SRE?
Standard event register	Changes in device status	–	*ESR?
Standard event enable register	Masks standard event register	*ESE	*ESE?
Extended event register	Changes in device status	–	STATus: EESR?
Extended event enable register	Masks extended event register	STATus: EESE	STATus: EESR?
Condition register	Current instrument status	–	STATus: CONDition?
Transition filter	Conditions that change the extended event register	STATus: FILTer<x>	STATus: FILTer<x>?
Output queue	Stores a response message to a query All query commands		
Error queue	Stores the error No. and message	–	STATus: ERRor?

Registers and Queues That Affect the Status Byte
Registers that affect the bits of the status byte are shown below.

Standard event register: Sets bit 5 (ESB) of the status byte to 1 or 0.
Output queue: Sets bit 4 (MAV) of the status byte to 1 or 0.
Extended event register: Sets bit 3 (EES) of the status byte to 1 or 0.
Error queue: Sets bit 2 (EAV) of the status byte to 1 or 0.

Enable Registers

Registers that are used to mask a bit so that the bit will not affect the status byte even when it is set to 1, are shown below.

Status byte: Mask the bits using the service request enable register.
Standard event register: Mask the bits using the standard event enable register.
Extended event register: Mask the bits using the extended event enable register.

Writing/Reading from Registers

The *ESE command is used to set the bits in the standard event register to 1's or 0's. The *ESE? command is used to query whether the bits in the standard event register are 1's or 0's. For details regarding these commands, see chapter 5.

6.2 Status Byte

Status Byte

- **Bits 0, 1, and 7**
  Not used (always 0)
- **Bit 2 EAV (Error Available)**
  Set to 1 when the error queue is not empty. In other words, this bit is set to 1 when an error occurs. See the page 6-5.
- **Bit 3 EES (Extend Event Summary Bit)**
  Set to 0 when the logical product of the extended event register and the corresponding enable register is 1. In other words, this bit is set to 1 when an event takes place inside the instrument. See the page 6-4.
- **Bit 4 MAV (Message Available)**
  Set to "1" when the output queue is not empty. In other words, this bit is set to 1 when there are data to be transmitted. See the page 6-5.
- **Bit 5 ESB (Event Summary Bit)**
  Set to 0 when the logical product of the standard event register and the corresponding enable register is 1. In other words, this bit is set to 1 when an event takes place inside the instrument. See the page 6-3.
- **Bit 6 RQS(Request Service)/MSS(Master Status Summary)**
  Set to 1 when the logical AND of the status byte excluding Bit 6 and the service request enable register is not 0. In other words, this bit is set to 1 when the instrument is requesting service from the controller.
  RQS is set to 1 when the MSS bit changes from 0 to 1, and cleared when serial polling is carried out or when the MSS bit changes to 0.

Bit Masking

To mask a bit in the status byte so that it does not cause an SRQ, set the corresponding bit of the service request enable register to 0.
For example, to mask bit 2 (EAV) so that service is not requested when an error occurs, set bit 2 of the service request enable register to 0. This can be done using the *SRE command. To query whether each bit of the service request enable register is 1 or 0, use *SRE?. For details on the *SRE command, see chapter 5.

Operation of the Status Byte

A service request is issued when bit 6 of the status byte becomes 1. Bit 6 is set to 1 when any of the other bits becomes a 1 (when the corresponding bit of the service request enable register is also set to 1).
For example, if an event occurs and the logical AND of the standard event register and the corresponding enable register becomes a 1, then bit 5 (ESB) is 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.
In addition, you can also check what type of event occurred by reading the contents of the status byte.

Reading from the Status Byte

The following two methods are provided for reading the status byte.

- **Inquiry using the *STB? query**
  Making an inquiry using the *STB? query sets bit 6 to MSS. This causes the MSS to be read. After completion of the read-out, none of the bits in the status byte will be cleared.
- **Serial polling**
  Execution of a serial polling changes bit 6 to RQS. This causes RQS to be read. After completion of the read-out, only RQS is cleared. It is not possible to read MSS using serial polling.

Clearing the Status Byte

No method is provided for forcibly clearing all the bits in the status byte. The bits that are cleared for each operation are shown below.

- **When a query is made using the *STB? command**
  No bits are cleared.
- **When serial polling is executed**
  Only the RQS bit is cleared.
- **When a *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.

6.3 Standard Event Register

Standard Event Register

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

- **Bit 7 PON (Power ON)**
  Set to 1 when the 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.
  Example 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.
  Example Received a command with a parameter outside the range or attempted to output a hard copy while waveform acquisition is in progress.
- **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.
- **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.
  Example 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 (see chapter 5) has been completed.

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. Refer to Chapter 4.
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 on the *ESE command, see chapter 5.

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).

Example

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 the register is read, it is cleared.

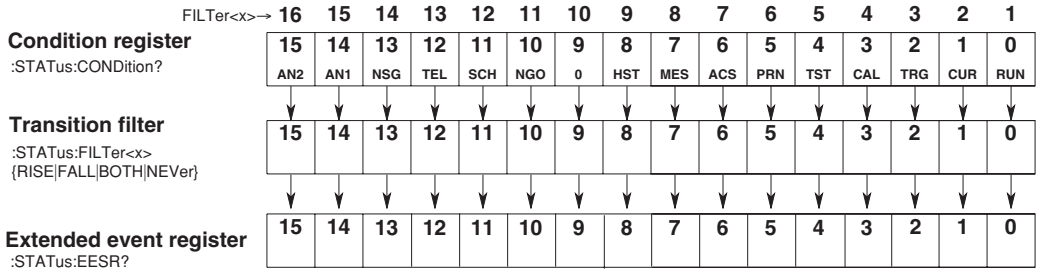
Clearing the Standard Event Register

The standard event register is cleared in the following three cases.

- When the contents of the standard event register are read using the *ESR command.
- When a *CLS command is received.
- When the instrument is power cycled.

6.4 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 while waveform acquisition is in progress.
Bit 1	CUR (Cursor)	Set to 1 during cursor measurement.
Bit 2	TRG (Awaiting trigger)	Set to 1 when waiting for a trigger.
Bit 3	CAL (Calibration)	Set to 1 while calibration is in progress.
Bit 4	TST (Testing)	Set to 1 while self-test is in progress.
Bit 5	PRN (Printing)	Set to 1 while the built-in printer is operating, while data is being output to an external printer (USB/network), or while screen image data is being saved.
Bit 6	ACS (Accessing)	Set to 1 while a storage drive is being accessed.
Bit 7	MES (Measuring)	Set to 1 when automated measurement of waveform parameters is in progress.
Bit 8	HST (History Search)	Set to 1 while history search is in progress.
Bit 10	NGO (Go/No-go)	Set to 1 while GO/NO-GO search is in progress.
Bit 11	SCH (Search)	Set to 1 while search is in progress.
Bit 12	TEL (Telecom Test)	Set to 1 while the telecom test is in progress.
Bit 13	NSG (N-Single)	Set to 1 while continuous acquisition is in progress when the trigger mode is set to single (N).
Bit 14	AN1 (Analysis1)	Set to 1 while Analysis 1 is in progress.
Bit 15	AN2 (Analysis2)	Set to 1 while Analysis 2 is in progress.

The transition filter parameters detect changes in the specified bit (numerical suffix, 1 to 16) of the condition register in the following manner and overwrite the extended event register.

RISE	The specified bit of the extended event register is set to 1 when the bit of the condition register changes from 1 to 0.
FALL	The specified bit of the extended event register is set to 1 when the bit of the condition register changes from 0 to 1.
BOTH	The bit of the extended event register is set to 1 when the bit of the condition register changes from 0 to 1 or from 1 to 0.
NEVer	Always 0.

6.5 Output Queue and Error Queue

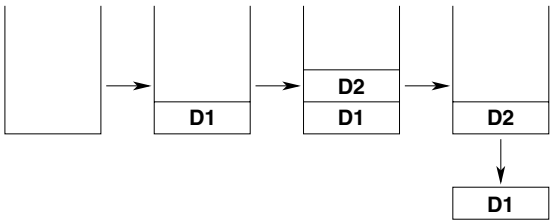
Output Queue

The output queue is provided to store response messages to queries. For example, if you send the WAVeform:SEND? command, which requests the output of acquired data, the data is stored in the output queue until it is read.

As shown below, data are stored in order and read from the oldest ones first. 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 a deadlock occurs (see page 5-2).
- When a device clear command (DCL or SDC) is received.
- When the instrument is power cycled.

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.



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 error number and message “113, “Undefined header”” are stored in the error queue when the error is displayed. The STATus:ERRor? query can be used to read the contents of the error queue. As with the output queue, the messages are read from the oldest ones first.

When the error queue overflows, the last message is replaced by the message “350, “Queue overflow”.”

The error queue is also cleared for the following cases:

- When a *CLS command is received.
- When the instrument is power cycled.

To see whether the error queue is empty or not, check bit 2 (EAV) of the status byte.

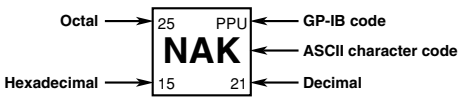
Appendix

Appendix 1 ASCII Character Codes

The following table shows the ASCII character codes.

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

Example



Appendix 2 Error Messages

This section describes the error messages related to communications.

- The messages can be displayed in English or Japanese on the DL9000. However, when the messages are read from a PC or other similar computers, the messages are displayed in English.
- If servicing is required, contact your nearest YOKOGAWA dealer for repairs.
- Only error messages related to communications are listed here. For other error messages, see *User's Manual IM 701310-01E*.
  - · Communication syntax error 100~199
  - · Communication execution error 200~299
  - · Model specific (other) 300~398
  - · Communication query error 400~499
  - · System error (communications) 399

  } Details given below.

Error in Communication Command (100-199)

Code	Messages	Corrective Action	Page
102	A syntax error. Syntax error.	Invalid syntax.	
103	<DATA SEPARATOR> is missing. Invalid separator.	Use a comma to separate the data.	
104	The <DATA> type is incorrect. Data type error.	Write using the correct data form.	
105	Device trigger function cannot be used. GET not allowed.	GET is not supported for responses to interface messages.	
108	There are too many <DATA>. Parameter not allowed.	Check the number of data points.	
109	Required <DATA> is missing. Missing parameter.	Enter the required data.	
111	<HEADER SEPARATOR> is missing. Header separator error.	Use a space to separate the header and data.	
112	<mnemonic> is too long. Program mnemonic too long.	Check the mnemonic (alphanumerical character string).	
113	No such command. Undefined header.	Check the header.	
114	The value of <HEADER> is not correct. Header suffix out of range.	Check the header.	
120	The mantissa of the value is missing. Numeric data error.	A number is required in the <NRf> form.	
123	The exponent is too large. Exponent too large.	Use a smaller exponent for <NR3> format.	
124	There are too many significant digits. Too many digits.	The value must be less than equal to 255 digits.	
128	Numeric data cannot be used. Numeric data not allowed.	Enter in a format other than <NRf> format.	
131	The unit is not correct. Invalid suffix.	Check the unit of the <Voltage>, <Time>, <Frequency>, and <Current>.	
134	The spelling of the unit is too long. Suffix too long.	Check the unit of the <Voltage>, <Time>, <Frequency>, and <Current>.	

Code	Messages	Corrective Action	Page
138	Units cannot be used. Suffix not allowed.	No units are allowed other than <Voltage>, <Time> <Frequency>, and <Current>.	
141	No such selection available. Invalid character data.	Select character data from the selections available in {...|...|...}.	
144	The spelling of <CHARACTER DATA> is too long. Character data too long.	Check the spelling of the character strings in {...|...|...}.	
148	<CHARACTER DATA> cannot be used. Character data not allowed.	Write in a data form other than {...|...|...}.	
150	There is no delimiter to the right of <STRING DATA>. String data error.	Enclose <String> in double quotation or single quotation marks.	
151	The contents of <STRING DATA> Invalid string data.	<String> is too long or contains characters which cannot be used.	
158	<STRING DATA> cannot be used. are inappropriate. String data not allowed.	Enter in a data format other than <Character string>.	
161	The data length of <BLOCK DATA> does not match. Invalid block data.	<Block data> is not allowed.	
168	<BLOCK DATA> cannot be used. Block data not allowed.	<Block data> is not allowed.	
171	There is an invalid character in the <EXPRESSION DATA>. Invalid expression.	Equations cannot be used.	
178	<EXPRESSION DATA> cannot be used. Expression data not allowed.	Equations cannot be used.	

Error in Communication Execution (200 to 299)

Code	Messages	Corrective Action	Page
221	There is a conflict in the setup information. Setting conflict.	Check the relevant settings.	
222	The data value is outside the range. Data out of range.	Check the range.	
223	The data byte length is too long. Too much data.	Check the length of the data.	
224	The data value is invalid. Illegal parameter value.	Check the range.	
241	The hardware is not implemented. Hardware missing.	Check the installed options.	
260	<EXPRESSION DATA> is not correct. Expression error.	Equations cannot be used.	

Error in Communication Query (400 to 499)

Code	Messages	Corrective Action	Page
410	Query transmission was aborted. Query INTERRUPTED.	Check transmission/reception order.	
420	There is no response that can be transmitted. Query UNTERMINATED.	Check transmission/reception order.	
430	Deadlock occurred. Aborting transmission. Query DEADLOCKED.	Limit the length of the program message including <PMT> to 1024 bytes or less.	
440	The order to request the response is not correct. Query UNTERMINATED after indefinite response.	Do not specify a query after the *IDN? or *OPT? command.	

Error in System Operation (399)

Code	Messages	Corrective Action	Page
399	Communication driver error. Fatal error in the communication driver.	Maintenance service is required.	

Warning (50)

Code	Messages	Corrective Action	Page
50	*OPC/? is in the middle of the message. *OPC/? exists in message.	Place the *OPC or *OPC? command at the end of the program message.	

Other Errors (350)

Code	Messages	Corrective Action	Page
350	Queue overflow.	Read the error queue.	

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 Correspondence Table of Measure Parameter Names

Name Displayed on Setup Screen	Name Used by Communication Commands	Name on Menu of the DL9000 the the DL9000 Screen When Displaying Measured Results
Max	MAXimum	Max
Min	MINimum	Min
High	HIGH	High
Low	LOW	Low
P-P	PTOPeak	P-P
Hi-Low	HILow	Hi-Low
+Over	POVershoot	+Over
-Over	NOVershoot	-Over
Rms	RMS	Rms
Mean	MEAN	Mean
Sdev	SDEViation	Sdev
IntegTY	TYINteg	ITY
C.Rms	CRMS	CRms
C.Mean	CMEan	CMean
C.Sdev	CSDeviation	CSdev
C.IntegTY	TYCInteg	CITY
Freq	FREQuency	Freq
1/Freq	PERFrequency	1/FR
Count	COUNt	Count
Burst	BURSt	Burst
+Width	PWIDth	+Width
-Width	NWIDth	-Width
Period	PERiod	Period
Duty	DUTYcycle	Duty
Rise	RISE	Rise
Fall	FALL	Fall
Delay	DELay	Dly

Appendix 4 Eye Pattern Parameter Name Table

Name Displayed on the Setup Menu of the DL9000 Screen	Name Used by Communication Commands	Name on the DL9000 Screen When Displaying Measured Results
Crossing %	PCROssing	Cross%
Eye Height	EHEight	EyeHi
Eye Width	EWIDth	EyeWid
Q Factor	QFACtor	QFact
Jitter	JITTer	Jitter
Duty Cycle Distriction %	PDUTycycle	DCDTime%
Vtop	VTOP	Vtop
Vbase	VBASe	Vbase
σ top	SDTop	σ top
σ base	SDBase	σ base
Tcrossing1	T1CRossing	Tcros1
Tcrossing2	T2CRossing	Tcros2
Vcrossing	VCRossing	Vcros
Ext Rate dB	DBERate	ERdB
Rise	RISE	Rise
Fall	FALL	Fall