

DLM3034HD, DLM3054HD
High Definition Oscilloscope
Communication Interface

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

Thank you for purchasing the DLM3034HD or DLM3054HD High Definition Oscilloscope. This Communication Interface User's Manual explains the following interface features and commands.

- USB interface
- Ethernet interface (including socket interface)
- GP-IB interface (Option)

To ensure correct use, please read this manual thoroughly before beginning operation. Keep this manual in a safe place for quick reference in the event that a question arises. The manuals for this instrument are listed on page iii. Please read all manuals.

Contact information of Yokogawa offices worldwide is provided on the following sheet.

Document No.	Description
PIM 113-01Z2	List of worldwide contacts

Notes

- The contents of this manual are subject to change without prior notice as a result of improvements to the product's performance and functionality. Refer to our website to view our latest manuals.
- The figures given in this manual may differ from those that actually appear on your screen.
- 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 is strictly prohibited.
- The TCP/IP software of this product and the documents concerning it have been developed/created by YOKOGAWA based on the BSD Networking Software, Release 1 that has been licensed from the Regents of the University of California.

Trademarks

- Microsoft, Windows, Windows 10, and Windows 11 are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe and Acrobat are either registered trademarks or trademarks of Adobe Systems Incorporated.
- DLM is a registered trademark of Yokogawa Electric Corporation.
- In this manual, the ® and TM symbols do not accompany their respective registered trademark or trademark names.
- Other company and product names are registered trademarks or trademarks of their respective holders.

About the USB interface and Ethernet interface

To use the USB communication features, your PC must have the following:

- DLM series Communication Library (TMCTL)
- USB driver for connecting the PC-DLM3000HD series to the PC

To use the Ethernet communication features, your PC must have the following:

- DLM series Communication Library (TMCTL)

To download the libraries and drivers listed above, go to the following website, and then browse to the download page.

<https://tmi.yokogawa.com/library/>

Sample programs

To download sample programs, go to the following website, and then browse to the download page.

<https://tmi.yokogawa.com/library/>

Revisions

- 1st Edition: October 2024

Manuals

The following manuals, including this one, are provided as manuals for this instrument. Please read all manuals.

Manuals included with the product

Manual Title	Manual No.	Description
DLM3034HD, DLM3054HD High Definition Oscilloscope Getting Started Guide	IM DLM3054HD-03EN	Explains the handling precautions, common operations, troubleshooting measures, and specifications of this instrument.
DLM3034HD, DLM3054HD High Definition Oscilloscope Operation Guide	IM DLM3054HD-04EN	Explains the basic operations of this instrument. Operations are described in steps from "Preparation" to "Displaying Waveforms," "Measuring Waveforms," and "Saving Screen Captures."
DLM3034HD, DLM3054HD High Definition Oscilloscope Request to Download Manuals	IM DLM3054HD-73Z2	Describes the manuals provided on the website.
DLM3034HD, DLM3054HD High Definition Oscilloscope	IM DLM3054HD-92Z1	Document for China
Safety Instruction Manual	IM 00C01C01-01Z1	Safety manual (European languages)

Manuals provided on the website

Download the following manuals from our website.

Manual Title	Manual No.	Description
DLM3034HD, DLM3054HD High Definition Oscilloscope Features Guide	IM DLM3054HD-01EN	Explains all the instrument's features other than the communication interface features.
DLM3034HD, DLM3054HD High Definition Oscilloscope User's Manual	IM DLM3054HD-02EN	Explains how to operate this instrument.
DLM3034HD, DLM3054HD High Definition Oscilloscope Communication Interface User's Manual	IM DLM3054HD-17EN	This document. Explains the functions of this instrument's communication interface, how to configure it, and the commands used to control this instrument from a PC through the interface.

For details on downloading manuals, see Request to Download Manuals (IM DLM3054HD-73Z2).

To view the PDF data, you need Adobe Acrobat Reader or a software application that can open PDF data.

The "EN," "Z1," and "Z2" in the manual numbers are the language codes.

Online help

The content similar to the Features Guide, IM DLM3054HD-01EN, is included in this instrument as a help file (some the content may be omitted). For instructions on how to use the help, see section 3.10 in the *Getting Started Guide*, IM DLM3054HD-03EN.

How to Use This Manual

Chapters in this manual

This manual contains six chapters and an appendix.

Chapter 1 USB Interface

Describes the features and specifications of the USB interface.

Chapter 2 Ethernet Interface

Describes the features and specifications of the Ethernet interface and the specifications of the socket interface.

Chapter 3 GP-IB Interface (Option)

Describes the features and specifications of the GP-IB interface.

Chapter 4 Programming Overview

Describes command syntax and other programming information.

Chapter 5 Commands

Describes every command individually.

Chapter 6 Status Reports

Describes the status byte, various registers, and queues.

Appendix

Provides reference material such as an ASCII character code table.

Index

Conventions Used in This Manual

Prefixes k and K

Prefixes k and K used before units are distinguished as follows:

k:	Denotes 1000.	Example: 100 kS/s (sample rate)
K:	Denotes 1024.	Example: 720 KB (file size)

Displayed characters

Bold characters in procedural explanations are used to indicate panel keys and soft keys that are used in the procedure and menu items that appear on the screen.

Notes and cautions

The notes and cautions in this manual are categorized using the following symbols.

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 attention to actions or conditions that could cause light injury to the user or cause 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 software.

Metasyntax

The following table contains the symbols that are used in the syntax discussed mainly in chapters 4 and 5. These symbols are referred to as BNF (Backus-Naur Form) symbols. For details on how to write data using these symbols, see pages 4-6 and 4-7.

Symbol	Description	Example	Example of Input
<>	A defined value	CHANnel<x> <x> = 1 to 4	CHANNEL2
{ }	Select an option in { }	COUPling {AC DC DC50}	COUPLING AC
	Exclusive OR		
[]	Can be omitted	CURSor [:TY] :TYPE	CURSor:TYPE

Contents

Manuals	iii
How to Use This Manual	iv
Conventions Used in This Manual	v

Chapter 1 USB Interface

1.1	Component Names and Functions.....	1-1
1.2	USB Interface Features and Specifications	1-2
1.3	USB Interface Connection.....	1-4
1.4	Configuring the Instrument's USB Settings	1-5

Chapter 2 Ethernet Interface

2.1	Component Names and Functions.....	2-1
2.2	Ethernet Interface Features and Specifications, Socket Interface Specifications	2-2
2.3	Ethernet Interface Connection	2-5
2.4	Configuring the Instrument's Network Settings	2-6

Chapter 3 GP-IB Interface (Option)

3.1	Component Names and Functions.....	3-1
3.2	GP-IB Interface Features and Specifications	3-2
3.3	GP-IB Interface Connection	3-4
3.4	Configuring the Instrument's GP-IB Settings	3-5
3.5	Responses to Interface Messages.....	3-6

Chapter 4 Programming Overview

4.1	Messages.....	4-1
4.2	Commands.....	4-4
4.3	Responses	4-7
4.4	Data.....	4-8
4.5	Synchronization with the Controller.....	4-12

Chapter 5 Commands

5.1	List of Commands	5-1
5.2	ACQuire Group	5-61
5.3	ANALysis Group.....	5-62
5.4	ASETup Group	5-82
5.5	CALibrate Group	5-83
5.6	CHANnel Group	5-84
5.7	CLEar Group	5-87
5.8	COMMunicate Group	5-88
5.9	CURSor Group	5-90
5.10	DISPlay Group	5-98
5.11	FFT Group.....	5-100
5.12	FILE Group.....	5-108
5.13	GONogo Group	5-112
5.14	HCOPy group.....	5-118
5.15	HISTory Group	5-120
5.16	IMAGe Group	5-126
5.17	INITialize Group	5-128

5.18	LOGic group.....	5-129
5.19	MATH Group	5-133
5.20	MEASure Group.....	5-139
5.21	RECall group.....	5-154
5.22	REFerence Group	5-155
5.23	SEARch Group.....	5-156
5.24	SERialbus group	5-162
5.25	SNAP Group	5-247
5.26	SSTart Group	5-248
5.27	STARt Group.....	5-249
5.28	STATus Group.....	5-250
5.29	STOP Group.....	5-251
5.30	STORe Group	5-252
5.31	SYSTem Group	5-253
5.32	TIMEbase Group	5-258
5.33	TRIGger Group	5-259
5.34	WAVEform Group	5-335
5.35	WPARameter Group	5-339
5.36	XY Group	5-343
5.37	ZOOM Group	5-346
5.38	Common Command Group.....	5-348

Chapter 6 Status Reports

6.1	About Status Reports.....	6-1
6.2	Status Byte.....	6-3
6.3	Standard Event Register	6-5
6.4	Extended Event Register	6-7
6.5	Output and Error Queues.....	6-8

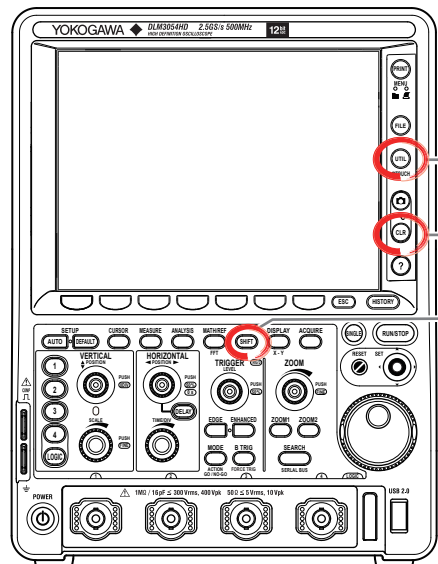
Appendix

Appendix 1	ASCII Character Codes.....	App-1
Appendix 2	Error Messages	App-2
Appendix 3	About the IEEE 488.2-1992 Standard	App-6

Index

1.1 Component Names and Functions

Front panel



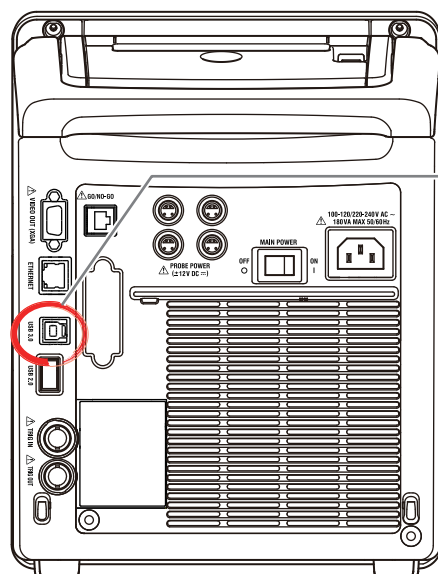
UTIL key

Press this key to select the USB interface.

SHIFT+CLR key

Press these keys to switch from remote mode, in which settings and operations are performed through remote commands, to local mode, in which operations can be performed using the

Rear panel



USB port for PCs

This port is for connecting the instrument to a controller (such as a PC) using a USB cable.

1.2 USB Interface Features and Specifications

USB interface features

Reception feature

You can use the reception feature to specify the same settings that you specify by using the front panel keys.

The instrument can receive output requests for measured and computed data, panel setup parameters, and error codes.

Transmission feature

The instrument can (1) transmit measured and computed data, (2) transmit panel setting data and the status byte, and (3) error codes when errors occur.

USB interface specifications

Electrical and mechanical	USB Rev. 3.0/2.0 compliant
Connector	Type B connector (receptacle)
Number of ports	1
Power supply	Self powered
System requirements	A PC running the English or Japanese version of Windows 10 (32 bit, 64 bit), or Windows 11 and a separate device driver

Data transfer rate

The following table contains approximations of how much time it takes for the instrument to transmit waveform data.

Model: DLM3054HD

Controller: PC (Intel (R) Xeon (R) E-2124 CPU @ 3.30 GHz 3.31 GHz), USB (Intel (R) USB 3.1 eXtensible Host Controller-1.10 (Microsoft)), OS (Windows 10 Enterprise 64 bit)

Programming language: Visual C++

Number of Data Points	Word Data
125000	9 ms
250000000	8996 ms

Switching between remote and local modes

Switching from local to remote mode

The instrument switches to remote mode when it is in local mode and receives a

`:COMMunicate:REMOte ON` command from the PC.

- “REMOTE” appears in the top center of the screen.
- All keys except the **SHIFT+CLR** key are disabled.
- Local mode settings are retained even when the instrument switches to remote mode.

Switching from remote to local mode

When the instrument is in remote mode and you press **SHIFT+CLR**, the instrument

switches to local mode. However, this does not work if the instrument has received a

`:COMMunicate:LOCKout ON` command from the PC. The instrument switches to local mode when it receives a `:COMMunicate:REMOte OFF` command from the PC, regardless of the local lockout state.

- The REMOTE indicator in the top center of the screen disappears.
- All keys are enabled.
- Remote mode settings are retained even when the instrument switches to local mode.

Note

You cannot use the USB interface simultaneously with other interfaces (GP-IB and Ethernet interfaces).

1.3 USB Interface Connection

Notes about connections

- Be sure to insert the USB cable connector firmly into the USB port.
- If you are connecting multiple devices by using a USB hub, connect the instrument to the USB hub port that is closest to the port that the controller is connected to.
- Do not connect a USB cable to the GO/NO-GO output terminal. Doing so may damage the instrument.
- Before connecting a PC to the USB port for PCs, ground the PC to the same electrical potential as the instrument.
- Do not connect a device other than a controller (e.g., PC) to the USB port for PCs.

1.4 Configuring the Instrument's USB Settings

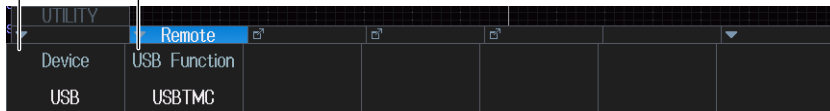
This section explains the communication interface settings for controlling the instrument remotely through a USB interface.

UTILITY Remote Control menu

Press **UTIL** and then the **Remote Control** soft key. The following menu appears.

Set Device to USB.

Set USB Function to USB TMC.

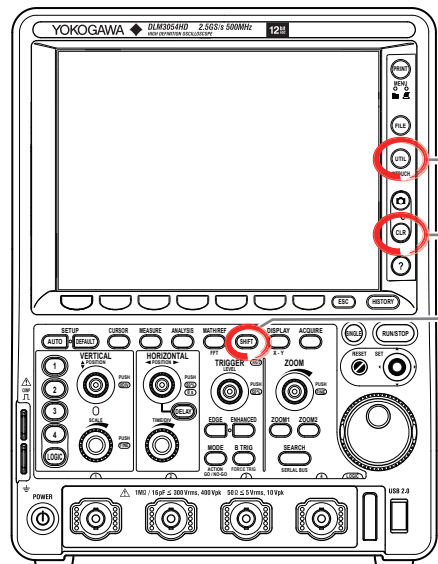


Note

- If you send commands simultaneously from another communication interface that is not selected, the instrument will not execute the commands properly.
- To remotely control the instrument through a USB port using communication commands, set USB Function, shown in the figure above, to USB TMC, and then carry out the following steps.
 - To activate the USB Function settings, you need to restart the instrument. Turn off the instrument power switch, wait at least 10 seconds, and then turn it back on.
 - Install the YOKOGAWA USB driver in your PC. For information about how to obtain the YOKOGAWA USB driver, contact your nearest YOKOGAWA dealer. You can also access the YOKOGAWA USB driver (YTUSB) download webpage and download the driver.
<https://tmi.yokogawa.com/jp/library/>
- Do not use USB drivers (or software) supplied by other companies.

2.1 Component Names and Functions

Front panel



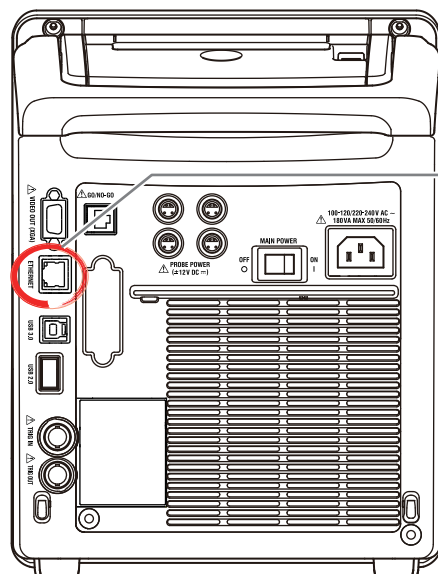
UTIL key

Press this key to select the Ethernet interface.

SHIFT+CLR key

Press these keys to switch from remote mode, in which settings and operations are performed through remote commands, to local mode, in which operations can be performed using the panel keys. These keys are disabled when local lockout (see page 2-3) has been activated by a controller.

Rear panel



Ethernet port

This port is for connecting the instrument to a controller (such as a PC) using an Ethernet cable.

2.2 Ethernet Interface Features and Specifications, Socket Interface Specifications

Ethernet interface features

Reception feature

You can use the reception feature to specify the same settings that you specify by using the front panel keys.

The instrument can receive output requests for measured and computed data, panel setup parameters, and error codes.

Transmission feature

The instrument can (1) transmit measured and computed data, (2) transmit panel setting data and the status byte, and (3) error codes when errors occur.

Ethernet interface specifications

Electrical and mechanical	IEEE802.3 compliant
Simultaneous connections	1
Protocol	VXI-11
System requirements	A PC running the English or Japanese version of Windows 10 (32 bit or 64 bit) or Windows 11

Data transfer rate

The following table contains approximations of how much time it takes for the instrument to transmit waveform data.

Model: DLM3054HD

Controller: PC (Intel(R) Core(TM) i5-7300U CPU @ 2.60 GHz 2.71 GHz), Ethernet (Intel(R)

Ethernet Connection (4) I219-LM), OS (Windows 10 Enterprise 64 bit)

Programming language: Visual C++

Number of Data Points	Word Data
125000	13 ms
250000000	18720 ms

Switching between remote and local modes

Switching from local to remote mode

The instrument switches to remote mode when it is in local mode and receives a `:COMMunicate:REMOte ON` command from the PC.

- “REMOTE” appears in the top center of the screen.
- All keys except the **SHIFT+CLR** key are disabled.
- Local mode settings are retained even when the instrument switches to remote mode.

Switching from remote to local mode

When the instrument is in remote mode and you press **SHIFT+CLR**, the instrument switches to local mode. However, this does not work if the instrument has received a `:COMMunicate:LOCKout ON` command from the PC. The instrument switches to local mode when it receives a `:COMMunicate:REMOte OFF` command from the PC, regardless of the local lockout state.

- The REMOTE indicator in the top center of the screen disappears.
- All keys are enabled.
- Remote mode settings are retained even when the instrument switches to local mode.

Note

You cannot use the Ethernet interface simultaneously with other interfaces (GP-IB and USB interfaces).

Timeout period

If the instrument is not accessed for a specified amount of time (set as a timeout period), the instrument closes the connection to the network. You can set the timeout value in the range of 1 s to 3600 s and Infinite. The default is Infinite.

For how to set the timeout period, see section 2.4, “Configuring the instrument Network Settings.”

Socket interface specifications

Electrical and mechanical	IEEE802.3 compliant
Simultaneous connections	1
Protocol	None
port number	10002/tcp
Terminator	LF (0Ah)
Connection timeout	None
System requirements	Windows 10 (32 bit, 64 bit), Windows 11
Restrictions	The maximum number of bytes of command that can be sent at once is 4 KB.

Socket communication sample program

```
#include "stdafx.h"
#include <stdio.h>
#include <winsock2.h>
#include <ws2tcpip.h>

#pragma comment (lib, "Ws2_32.lib")

int _tmain(int argc, _TCHAR* argv[])
{
    /* Socket */
    int dstSocket;

    /* sockaddr_in structure */
    struct sockaddr_in dstAddr;

    /* Receive buffer */
    char buffer[200];

    /* ***** */
    /* Windows Socket API initialization */
    WSADATA data;
    WSAStartup(MAKEWORD(2,0), &data);

    /* Set the sockaddr_in structure */
    memset(&dstAddr, 0, sizeof(dstAddr));
    dstAddr.sin_port = htons(10002);
    dstAddr.sin_family = AF_INET;
    dstAddr.sin_addr.s_addr = inet_addr("192.168.0.1");

    /* Socket generation */
    dstSocket = socket(AF_INET, SOCK_STREAM, 0);

    /* Connection */
    connect(dstSocket, (struct sockaddr *) &dstAddr, sizeof(dstAddr));

    /* Packet transmission */
    send(dstSocket, "*IDN?¥n", 6, 0);

    /* Packet reception */
    recv(dstSocket, buffer, 200, 0);

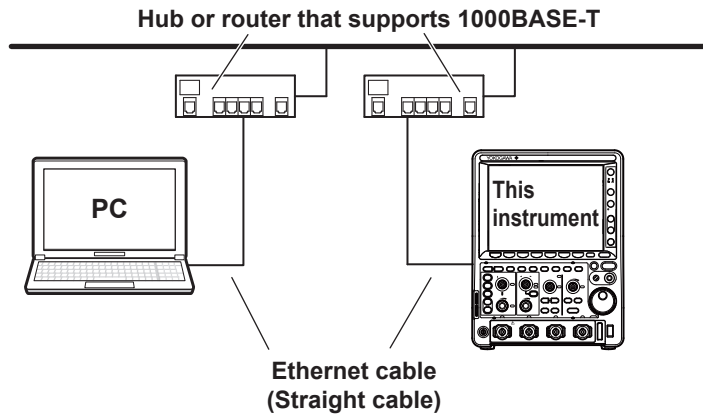
    /* Windows Socket API closure */
    closesocket(dstSocket);
    WSACleanup();

    return 0;
}
```

2.3 Ethernet Interface Connection

Connection procedure

Connect an Ethernet cable that is connected to a hub or other network device to the Ethernet port on the instrument's rear panel.



Notes about connections

- To connect the instrument to a PC, be sure to use straight cables and to connect through a hub or router. Proper operation is not guaranteed for a one-to-one connection using a crossover cable.
- Use a network cable appropriate for the data rate of your network.

Note

For details on how to connect the instrument to a network, see section 18.1, "Connecting the Instrument to a Network" in the *User's Manual*, IM DLM3054HD-02EN.

2.4 Configuring the Instrument's Network Settings

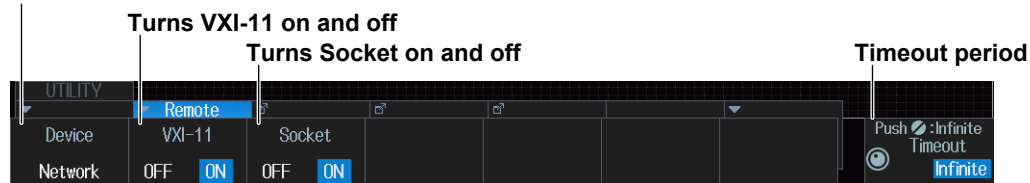
This section explains the following setting for remotely controlling the instrument via the Ethernet interface:

- Communication interface
- Turning the network connecting on or off (VXI-11 and Socket)
- Timeout period of the network

UTILITY Remote Control menu

Press **UTIL** and then the **Remote Control** soft key. The following menu appears.

Set Device to Network.



Note

- If you send commands simultaneously from another communication interface that is not selected, the instrument will not execute the commands properly.
- If VXI-11 or Socket is on, the Ethernet interface or the socket interface can be used. We recommend you turning them off if you are not using these features.

TCP/IP

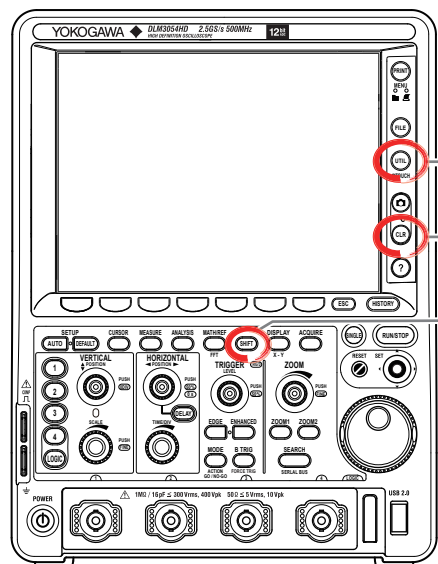
To use the Ethernet interface, you must specify the following TCP/IP settings.

- The IP address
- Subnet mask
- Default gateway

For instructions on how to specify these settings, see section 18.2, “Configuring TCP/IP Settings” in the *User’s Manual*, IM DLM3054HD-02EN.

3.1 Component Names and Functions

Front panel



UTIL key

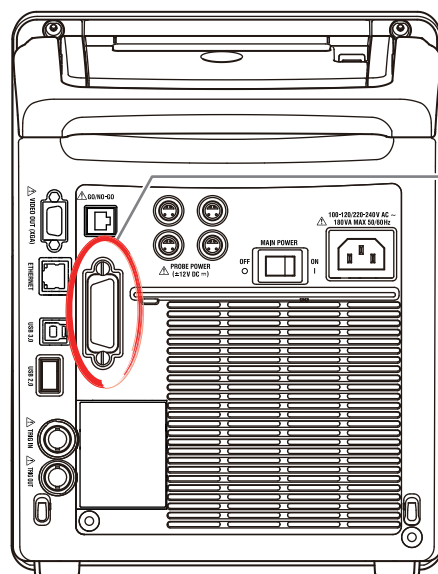
Press this key to select the GP-IB interface.

SHIFT+CLR key

Press these keys to switch from remote mode, in which settings and operations are performed through remote commands, to local mode, in which operations can be performed using the panel keys.

These keys are disabled when local lockout (see page 3-3) has been activated by a controller.

Side panel



GP-IB port

This port is for connecting the instrument to a controller (such as a PC) using a GP-IB cable.

3.2 GP-IB Interface Features and Specifications

GP-IB interface features

Listener capabilities

- Allows you to specify the same instrument settings that you can using the front panel keys. You cannot turn the power on and off or change communication settings.
- Receive output requests for measured and computed data, panel setting data, and error codes.
- Receives status report commands and other commands.

Talker capabilities

The instrument can (1) transmit measured and computed data, (2) transmit panel setting data and the status byte, and (3) error codes when errors occur.

Note

Talk-only, listen-only, and controller capabilities are not available.

GP-IB interface specifications

Electrical and mechanical	IEEE St'd 488-1978
Functional specifications	See the table below.
Protocol	IEEE St'd 488.2-1992
Code	ISO (ASCII) codes
Mode	Addressable mode
Address setup	Set the address in the range of 0 to 30 on the GP-IB setup screen of the UTILITY menu.
Clearing remote mode	Clear remote mode by pressing SHIFT+CLR key except when Local Lockout is enabled from 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
Electric characteristics	E1	Open collector

Data transfer rate

The following table contains approximations of how much time it takes for the instrument to transmit waveform data.

Model: DLM3054HD

Controller: PC (Intel (R) Xeon (R) E-2124 CPU @ 3.30 GHz 3.31 GHz), GP-IB (NI PCI-GPIB), OS
(Windows 10 Enterprise 64 bit)

Programming language: Visual C++

Number of Data Points	Word Data
125000	350 ms
250000000	710163 ms

Switching between remote and local modes

Switching from local to remote mode

When the instrument is in Local mode and it receives a REN (Remote Enable) message from the PC, the instrument switches to Remote mode.

- “REMOTE” appears in the top center of the screen.
- All keys except the **SHIFT+CLR** key are disabled.
- Local mode settings are retained even when the instrument switches to remote mode.

Switching from remote to local mode

When the instrument is in remote mode and you press **SHIFT+CLR**, the instrument switches to local mode. These keys are disabled when local lockout (see page 3-6) has been activated by a controller.

- The REMOTE indicator in the top center of the screen disappears.
- All keys are enabled.
- Remote mode settings are retained even when the instrument switches to local mode.

Note

You cannot use the GP-IB interface simultaneously with other interfaces (USB and Ethernet interfaces).

3.3 GP-IB Interface Connection

CAUTION

Be sure to turn off the PC and the instrument when connecting or removing communication cables. Otherwise, erroneous operation may result, or the internal circuitry may break.

French

ATTENTION

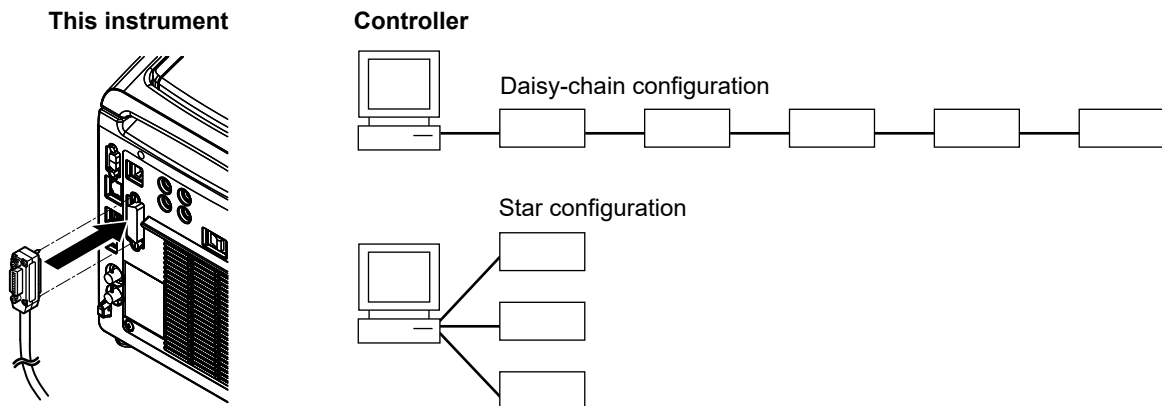
Veillez à mettre le PC et l'oscilloscope instrument hors tension lorsque vous branchez ou débranchez les câbles de communication, car cela risquerait de provoquer des dysfonctionnements ou des courts-circuits internes.

GP-IB cable

The instrument is equipped with a IEEE St'd 488-1978 24-pin GP-IB connector. Use GP-IB cables that complies with IEEE St'd 488-1978.

Connection procedure

Connect the cable as shown below. To connect multiple devices, use a daisy-chain or star configuration. You can also mix these configurations. Loop configuration is not allowed.



Notes about connections

- Securely fasten the GP-IB cable connector screws.
- On the PC end, use a GP-IB board (or card) made by National Instruments. For more details, see section 3.2.
- The instrument may not operate properly if the instrument is connected to the PC through converters (such as a GP-IB to USB converter). For more details, contact your nearest YOKOGAWA dealer.
- Several 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, you must assign a unique address to each device.
- Use cables that are 2 m or shorter in length to connect devices.
- Keep the total length of the cables under 20 m.
- When devices are communicating, have at least two-thirds of the devices on the bus turned on.

3.4 Configuring the Instrument's GP-IB Settings

This section explains the settings listed below. You must configure these settings when controlling the instrument remotely through a GP-IB interface.

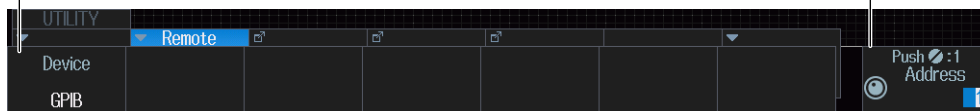
- Communication interface
- GP-IB address

UTILITY Remote Control menu

Press **UTIL** and then the **Remote Control** soft key. The following menu appears.

Set Device to GPIB.

GP-IB address



Note

- If you send commands simultaneously from another communication interface that is not selected, the instrument will not execute the commands properly.
- When the controller is communicating with the instrument or with other devices through GP-IB, do not change the address.
- Each device that is connected in a GP-IB system has its own unique address. This address is used to distinguish between different devices. Therefore, you must assign a unique address to the instrument when connecting it to a PC or other device.

3.5 Responses to Interface Messages

Responses to interface messages

Responses to uni-line messages

IFC (Interface Clear)

Clears the talker and listener functions. Stops data transmission if it is in progress.

REN (Remote Enable)

Switches between the remote and local modes.

IDY (Identify) is not supported.

Responses to multi-line messages (address commands)

GTL (Go To Local)

Switches the instrument to local mode.

SDC (Selected Device Clear)

- Clears the program message (command) being received and the output queue (see section 6.5).
- Discards *OPC and *OPC? commands that are being executed.
- Immediately aborts *WAI and COMMunicate:WAIT commands.

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

Responses to multi-line messages (universal commands)

LLO (Local Lockout)

Disables the **SHIFT+CLR** key on the front panel to prohibit switching to the local mode.

DCL (Device Clear)

Performs the same operation as SDC.

SPE (Serial Poll Enable)

Sets the talker function on all devices on the bus to serial polling mode. The controller will poll each device one by one.

SPD (Serial Poll Disable)

Clears the talker function's serial poll mode on all devices on the bus.

PPU (Parallel Poll Unconfigure) is not supported.

What are interface messages?

Interface messages are commands that a controller transmits. They are also referred to as interface commands or bus commands. They are classified as follows:

Uni-line messages

Uni-line messages are sent over a single control line. The following three messages are available.

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

Multi-line messages

Multi-line messages are sent over eight data lines. The messages are grouped as follows:

Address commands

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

Commands available to a device designated as a listener

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

Commands available to a device designated as a talker

- TCT (Take Control)

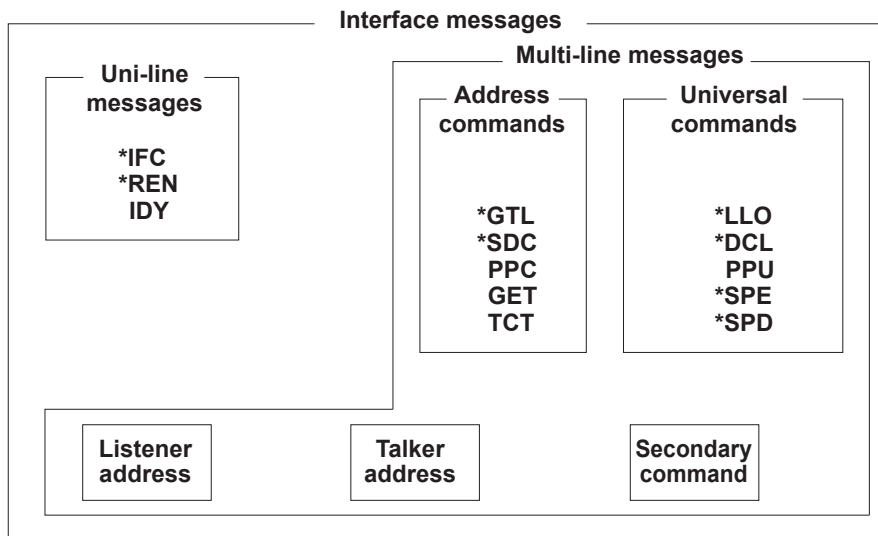
Universal commands

Universal commands are available to all devices regardless of their listener or talker designation. The following five commands are available.

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

3.5 Responses to Interface Messages

There are other interface messages: listener-address, talker-address, and secondary commands.



The instrument supports interface messages marked with an asterisk.

Note

Difference 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.

4.1 Messages

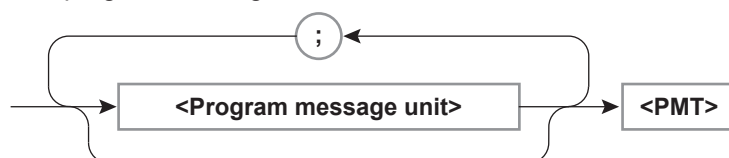
Messages

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 from the instrument back to the controller are called response messages.

If a program message contains a command that requests a response (a query), the instrument returns a response message upon receiving the program message. The instrument returns a single response message in response to a single program message.

Program messages

The program message format is shown below.



<Program message unit>

A program message consists of one or more program message units. Each unit corresponds to one command. The instrument executes the commands in the order that they are received. Separate each program message unit with a semicolon.

For details on the program message syntax, see the next section.

Example `:ACquire:MODE NORMal;RESolution 1<PMT>`

Unit
Unit

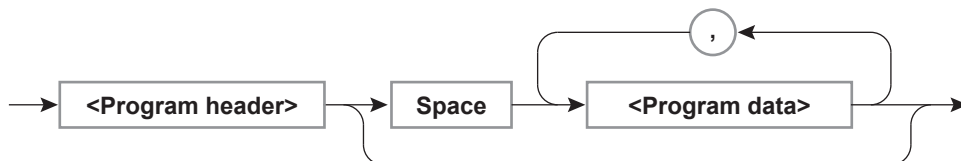
<PMT>

This 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 with the END message is the last data byte of the program message.)
NL^EOM	NL with an END message attached. (NL is not included in the program message.)

Program message unit syntax

The program message unit syntax is shown below.



<Program header>

The program header indicates the command type. For details, see section 4.2.

<Program data>

Attach program data if there are conditions that are required to execute a command. Separate the program data from the header with a space (ASCII code 20H). If there are multiple sets of program data, separate each set with a comma.

For details, see section 4.3.

Example

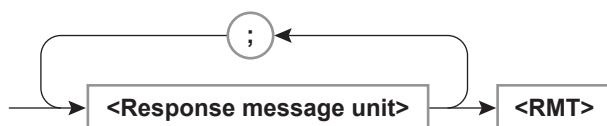
```

:ACQuire:MODE NORMAl<PMT>
  Header      Data
  
```

The example shows the command ":ACQuire:MODE NORMAl<PMT>". A bracket under "ACQuire:MODE" is labeled "Header", and a bracket under "NORMAl<PMT>" is labeled "Data".

Response messages

The response message syntax is as follows:



<Response message unit>

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

Separate each response message unit with a semicolon.

For details on the response message syntax, see the next page.

Example

```

:ACQUIRE:MODE NORMAL;RESolution 1<RMT>
  Unit              Unit
  
```

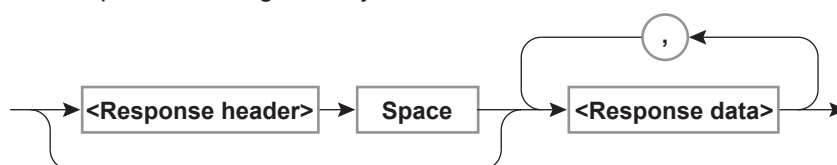
The example shows the command ":ACQUIRE:MODE NORMAL;RESolution 1<RMT>". A bracket under "ACQUIRE:MODE NORMAL" is labeled "Unit", and a bracket under "RESolution 1" is labeled "Unit".

<RMT>

RMT stands for "response message terminator." The response message terminator is NL^EOM.

Response message unit syntax

The response message unit syntax is as follows:



<Response header>	A response header sometimes precedes the response data. Separate the data from the header with a space. For details, see section 4.3.
<Response data>	Response data contains the content of the response. If there are multiple sets of response data, separate each set with a comma. For details, see section 4.3.

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

If there are multiple queries in a program message, responses are returned in the same order that the queries were received in. 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.

Notes on sending and receiving messages

- If the controller sends a program message that does not contain a query, the controller can send the next program message at any time.
- If the controller sends a program message that contains a query, the controller must finish receiving the response message before it can send the next program message. If the controller sends the next program message before receiving the response message in its entirety, an error will occur. A response message that is not received in its entirety will be discarded.
- If the controller tries to receive a response message when there is none, an error will occur. If the controller tries to receive a response message before the transmission of the program message is complete, an error will occur.
- If the controller sends a program message containing multiple message units, but the message contains incomplete units, the instrument will try to execute the ones that are believed to be complete. However, these attempts may not always be successful. In addition, if such a message contains queries, the instrument may not necessary return responses.

Deadlock

The instrument can store at least 1024 bytes of messages in its transmit and receive buffers (the number of available bytes varies depending on the operating conditions). If both the transmit and receive buffers become full at the same time, the instrument will no longer be able to operate. This condition is called a deadlock. If this happens, you can resume operation by discarding response messages.

Deadlock will not occur if the program message (including the <PMT>) is kept below 1024 bytes. Program messages that do not contain queries never cause deadlocks.

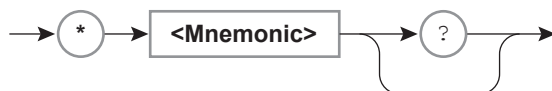
4.2 Commands

Commands

There are three types of commands (program headers) that a controller may send to the instrument. The commands differ in their program header formats.

Common command header

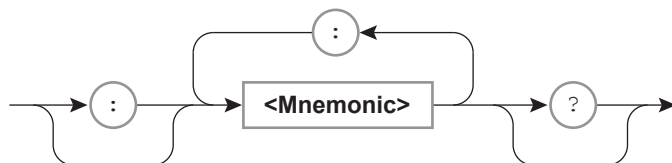
Commands that are defined in USBTMC-USB488 are called common commands. The header format of a common command is shown below. Be sure to include an asterisk (*) at the beginning of a common command.



Common command example *CLS

Compound header

Other commands that are specific to the instrument are classified and arranged in a hierarchy according to their functions. The compound header syntax is shown below. Be sure to use a colon to specify a lower hierarchical level.



Compound header example: :ACQuire:MODE

Simple header

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



Simple header example: START

Note

A <mnemonic> is an alphanumeric character string.

Concatenating commands

Command groups

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:RLENgth

Concatenating commands of the same group

The instrument stores the hierarchical level of the command that is currently being executed and processes the next command on the assumption that it belongs to the same level. Therefore, the common header section can be omitted for commands that belong to the same group.

Example :ACQuire:MODE NORMal;SAMPling REAL<PMT>

Concatenating commands of different groups

If the subsequent command does not belong to the same group, place a colon in front of the header (this colon cannot be omitted).

Example :ACQuire:MODE NORMal;;DISPlay:FORMat SINGLE<PMT>

Concatenating simple headers

If a simple header follows another command, place a colon in front of the simple header (this colon cannot be omitted).

Example :ACQuire:MODE NORMal;;START<PMT>

Concatenating common commands

Common commands that are defined in the USBTMC-USB488 are independent of hierarchy. A colon is not needed before a common command.

Example :ACQuire:MODE NORMal;*CLS;SAMPling REAL<PMT>

Separating commands with <PMT>

If you separate two commands with a terminator, two program messages will be sent. 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:SAMPling REAL<PMT>

Upper-level query

An upper-level query is a query that is made by appending a question mark to a command higher in the group. The controller can receive all of the settings in a group collectively by executing a highest-level query. Some upper-level queries of a group, which may be comprised of more than three hierarchical levels, can cause the instrument to transmit all the lower level settings.

Example :ACQUIRE? -> :ACQUIRE:AVERAGE:COUNT 2;ACQUIRE:COUNT INFINITY;
 MODE NORMAL;RESOLUTION 0;RLENGTH 125000;SAMPLING INTERPOLATE

The response to an upper-level query can be sent back to the instrument as a program message. This enables the settings that were present when the upper-level query was made to be reproduced later on. However, some upper-level queries do not return setup parameters that are not currently in use. Exercise caution because not all of a group's information is necessarily returned in a response.

Header interpretation rules

The instrument interprets the header that it receives according to the rules below.

- Mnemonics are not case sensitive.

Example CURSor can be written as cursor or Cursor.

- The lower-case characters can be omitted.

Example CURSor can be written as CURSO or CURS.

- The question mark at the end of a header indicates that it is a query. You cannot omit the question mark.

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 you write CHAN for CHANnel<x>, CHANnel1 is specified.

- Parts of commands and parameters enclosed in square brackets ([]) can be omitted.

Example TRIGger[:ATRigger]:SIMPlE:LEVel can be written as TRIG:SIMP:LEV.

However, the last section enclosed in brackets cannot be omitted in an upper-level query.

Example TRIGger? and TRIGger:SIMPle? are different queries.

4.3 Responses

Responses

When the controller sends a query with a question mark, the instrument returns a response message to the query. The instrument returns response messages in one of the following two forms.

Response consisting of a header and data

Responses that can be used as program messages without any changes are returned with command headers attached.

Example :ACQuire:MODE?<PMT>-> :ACQuire:MODE NORMAL<RMT>

Response only consisting of data

Responses that cannot be used as program messages unless changes are made (query-only commands) are returned without headers. However, there are query-only commands whose responses the instrument will attach headers to.

Example :MEASure:CHANnel1:PTOPeak:VALue?<PMT> -> 10.0E+00<RMT>

If you want the instrument to return responses without headers

You can configure the instrument so that even responses that have both headers and data are returned without headers. Use the `COMMunicate:HEADer` command for this purpose.

Abbreviated form

The instrument normally returns response headers with the lower-case section removed. You can configure the instrument so that full headers are returned. Use the `COMMunicate:HEADer` command for this purpose. The sections enclosed in square brackets ([]) are also omitted in the abbreviated form.

4.4 Data

Data contains conditions and values that are written after the header. A space separates the data from the header. Data is grouped as follows:

Data	Description
<Decimal>	A value expressed in decimal notation Example: Probe attenuation for CH1 -> CHANnel1:PROBe 100
<Voltage> <Time> <Frequency> <Current>	A physical value Example: Time-axis range -> TIMEbase:TDIV 1US
<Register>	A register value expressed as binary, octal, decimal, or hexadecimal Example: Extended event register value -> STATUS:EESe #HFE
<Character Data>	Predefined character string (mnemonic). Select from the available strings in braces. Example: Select the input coupling of CH1 -> CHANnel1:COUPling {AC DC DC50}
<Boolean>	Indicates on and off. Specify ON, OFF, or a value. Example: Turn on the CH1 display -> CHANnel1:DISPlay ON
<String data>	User-defined string Example: Comment attached to screen data output -> FILE:SAVE:COMMeNt
<Filename>	Indicates a file name. Example: Save file name -> FILE:SAVE:WAVEform:NAME "CASE1"
<Block data>	Data that contains 8-bit values Example: Response to acquired waveform data -> #800000010ABCDEFGH1J

<Decimal>

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

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

- The instrument can receive decimal values that are sent from the controller in any of the forms <NR1> to <NR3>. This is expressed as <NRf>.
- The instrument returns a response to the controller in one of the forms from <NR1> to <NR3> depending on the query. The same form is used regardless of the size of the value.
- For the <NR3> form, the plus sign after the "E" can be omitted. You cannot omit the minus sign.
- If a value outside the range is entered, the value is adjusted to the closest value within the range.
- If a value has more significant digits than are available, the value will be rounded.

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

<Voltage>, <Time>, <Frequency>, and <Current> indicate decimal values that have physical significance. A <Multiplier> or <Unit> can be attached to the <NRf> form that was described earlier. Use one of the following syntaxes.

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

Multipliers and units that you can use are indicated in the following table.

<Multiplier>			<Unit>		
Symbol	Word	Multiplier	Symbol	Word	Multiplier
EX	Exa	10 ¹⁸	V	Volt	Voltage
PE	Peta	10 ¹⁵	S	Second	Time
T	Tera	10 ¹²	HZ	Hertz	Frequency
G	Giga	10 ⁹	MHZ	Megahertz	Frequency
MA	Mega	10 ⁶	A	Ampere	Current
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	Atto	10 ⁻¹⁸			

- <Multiplier> and <Unit> are not case sensitive.
- “U” is used to indicate micro (“μ”).
- “MA” is used for Mega to distinguish it from Milli. Megahertz, which is expressed as “MHZ,” is an exception. Therefore, “M (Milli)” 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. This is used when each bit of the value has a particular meaning. Use one of the following syntaxes.

Syntax	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 in the <NR1> form.

<Character data>

<Character data> is a predefined character string (mnemonics). It is mainly used to indicate options and is chosen from the character strings given in { }. The data interpretation rules are the same as those described in “Header Interpretation Rules” on page 4-6.

Syntax	Example
{AC DC DC50}	AC

- As with the header, the `COMMunicate:VERBoSe` command can be used to select whether response messages are returned in the full form or in the abbreviated form.
- The `COMMunicate:HEADer` setting does not affect <Character data>.

<Boolean>

<Boolean> is data that indicates ON or OFF. Use one of the following syntaxes.

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

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

<String data>

<String data> is not a predefined character string like <character data>. It can be any character string. The character string must be enclosed in single quotation marks (') or double quotation marks (").

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

- If a character string contains a double quotation mark ("), the double quotation mark is expressed as two consecutive quotation marks ("""). This rule also applies to single quotation marks.
- A response message is always enclosed in double quotation marks (").
- <String data> is any character string. Therefore, the instrument assumes that the remaining program message units are part of the character string if no closing single (') or double quotation mark (") is encountered. As a result, no error is detected if a quotation mark is omitted.
- For information about the number of characters in a file name, see the Features Guide.

<Block data>

<Block data> is any 8-bit data. It is only used in response messages on the instrument. The syntax is as follows:

Syntax	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.
- <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 included in the 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 and sequential. Most commands are sequential commands.

Note

Overlap commands are indicated as such in chapter 5. All other commands are sequential commands.

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

If you specify V/div and send the next program message to query for the result, the instrument always returns the most recent setting (5 V in this case).

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

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

On the other hand, for example, if you execute a file load, query for the V/div value, and send the above program message, CHANnel1:VDIV? is executed before the loading of the file is completed, and the returned V/div value is the value before the file is loaded. Overlapping refers to the act of executing the next command before the processing of the current command is completed, such as in the command FILE:LOAD:SETup:EXECute "CASE1".

With overlap commands, the execution of the next command may start before the execution of the previous command is completed. You can prevent overlapping by using the following methods.

Synchronizing to overlap commands

You can use any of the following methods to synchronize to overlap commands and prevent overlapping.

Using the *WAI command

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

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

In the above example, the COMMunicate:OPSE command is used to select which command to apply *WAI to. Here, it is applied to the media access command. *WAI is executed before CHANnel1:VDIV?, so CHANnel1:VDIV? is not executed until the file loading is completed.

Using the COMMunicate:OVERlap command

The COMMunicate:OVERlap command enables (or disables) overlapping.

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

In the above example, COMMunicate:OVERlap #HFFBF enables overlapping for commands other than media access. Because overlapping of file loading is disabled, FILE:LOAD:SETup:EXECute "CASE1" operates in the same way as a sequential command. Thus, CHANnel1:VDIV? is not executed until the file loading is completed.

Using the *OPC command

The *OPC command sets the OPC bit, which is bit 0 in the standard event register (see section 6.3), to 1 when the overlapping is completed.

```
: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>
```

In the above example, the COMMunicate:OPSE command is used to select which command to apply *OPC to. Here, it is applied to the media access command. *ESE 1 and *SRE 32 indicate that a service request is only generated when the OPC bit becomes 1. *ESR? clears the standard event register. Thus, CHANnel1:VDIV? is not executed until the a service request is generated.

Using the *OPC? query

The *OPC? query generates a response when an overlapping operation is completed.

```
:COMMunicate:OPSE #H0040;;FILE:LOAD:SETup:EXECute "CASE1";
*OPC?<PMT>
(Read the response to *OPC?)
:CHANnel1:VDIV?<PMT>
```

In the above example, the COMMunicate:OPSE command is used to select which command to apply *OPC? to. Here, it is applied to the media access command. Because *OPC? does not generate a response until the overlapping operation is completed, the file loading will have been completed by the time the response to *OPC? is read.

Synchronizing to overlap commands (normal commands)

Even with sequential commands, synchronization with non-communication events such as triggers is sometimes required.

```
:ACQuire:COUNT 1;:TRIGger:MODE NORMAl<PMT>  
:WAVeform:SEND?<PMT>
```

For example, if you set the waveform acquisition count and send the above program message to query for waveform data acquired with the trigger mode set to normal, the `WAVeform:SEND?` command may be executed regardless of whether or not the acquisition has been completed and may result in a command execution error.

If this happens, you must use one of the following methods to synchronize to the end of waveform acquisition.

Using the `STATus:CONDition?` query

`STATus:CONDition?` is used to query for the contents of the condition register (see page 6-7).

```
:ACQuire:COUNT 1;:TRIGger:MODE NORMAl<PMT>  
:STATus:CONDition?<PMT>  
(Read the response. If bit 0 is 1, return to the previous command.)  
:WAVeform:SEND?<PMT>
```

In the above example, you can determine whether or not waveform acquisition is in progress by reading bit 0 in the condition register. If the bit is 1, waveform acquisition is in progress. If the bit is 0, waveform acquisition is not in progress.

`:SOURce:SEND?` is not executed until bit 0 in the condition register becomes 0.

Using the extended event register

(When service requests (SRQs) are available)

The changes in the condition register can be reflected in the extended event register (see page 6-7).

```
:STATus:FILTER1 FALL;:STATus:EES 1;EESR?;*SRE 8<PMT>
:ACQuire:COUNT 1;:TRIGger:MODE NORMAl<PMT>
(Read the response to STATus:EESR?)
(Wait for a service request)
:WAVEform:SEND?<PMT>
```

In the above example, the `STATus:FILTER1 FALL` command sets the transition filter so that bit 0 in the extended event (FILTER1) is set to 1 when bit 0 in the condition register changes from 1 to 0.

The `STATus:EES 1` command is used to only change the status byte based on bit 0 in the extended event register.

The `STATus:EESR?` command is used to clear the extended event register.

The `*SRE 8` command is used to generate service requests based only on the changes in the extended event register bits. Thus, `WAVEform:SEND?` is not executed until a service request is generated.

Using the COMMunicate:WAIT command

The `COMMunicate:WAIT` command is used to wait for a specific event to occur.

```
:STATus:FILTER1 FALL;:STATus:EESR?<PMT>
:ACQuire:COUNT 1;:TRIGger:MODE NORMAl<PMT>
(Read the response to STATus:EESR?)
:COMMunicate:WAIT 1;:WAVEform:SEND?<PMT>
```

In the above example, the meanings of `STATus:FILTER1 FALL` and `STATus:EESR?` are the same as in the extended event register case described earlier.

The `COMMunicate:WAIT 1` command specifies that the program will wait for bit 0 (RUN) in the extended event register to be set to 1. `WAVEform:SEND?` is not executed until bit 0 in the extended event register becomes 1.

5.1 List of Commands

Command	Function	Page
ACQUIRE Group		
:ACQUIRE?	Queries all waveform acquisition settings.	5-61
:ACQUIRE		
:AVERAGE?	Queries all settings related to averaging and waveform acquisition counts.	5-61
:AVERAGE		
:COUNT	Sets or queries the attenuation constant of exponential averaging or the average count of simple averaging when in averaging mode.	5-61
:COUNT	Sets or queries the number of waveform acquisitions to perform in normal, envelope, and averaging modes.	5-61
:MODE	Sets or queries the waveform acquisition mode.	5-61
:RESOLUTION	Sets or queries the high resolution mode on/off state.	5-61
:RLENGTH	Sets or queries the record length.	5-61
:SAMPLING	Sets or queries the sampling mode.	5-61
ANALYSIS Group		
:ANALYSIS?	Queries all analysis feature settings.	5-62
:ANALYSIS		
:AHISTOGRAM<x>?	Queries all settings related to the histogram of the analysis feature.	5-62
:AHISTOGRAM<x>		
:DISPLAY	Sets or queries the waveform histogram display on/off state.	5-62
:HORIZONTAL	Sets or queries the horizontal range of the waveform histogram.	5-62
:MEASURE?	Queries all settings related to the waveform histogram measurement (including the mode on/off state).	5-62
:MEASURE		
:MODE	Sets or queries the waveform histogram measurement mode.	5-62
:PARAMETER?	Queries all settings related to the waveform histogram measurement when the mode is set to Param.	5-62
:PARAMETER		
:ALL	Collectively turns on or off all the measurement items of the waveform histogram.	5-62
:<Parameter>?	Queries all settings related to the specified measurement item of the waveform histogram.	5-62
:<Parameter>		
:STATE	Sets or queries the on/off state of the specified measurement item of the waveform histogram.	5-63
:VALUE?	Queries the automatically measured value of the specified measurement item of the waveform histogram.	5-63
:HPOSITION<y>	Sets or queries the horizontal position of Cursor1 or Cursor2 of the waveform histogram.	5-63
:VPOSITION<y>	Sets or queries the vertical position of Cursor1 or Cursor2 of the waveform histogram.	5-63
:MODE	Sets or queries the target axis of the waveform histogram.	5-63
:RANGE	Sets or queries the measurement source window of the waveform histogram.	5-63
:TRACE	Sets or queries the source waveform of the waveform histogram.	5-63
:VERTICAL	Sets or queries the vertical range of the waveform histogram.	5-64
:PANALYZE?	Queries all power supply analysis settings.	5-64
:PANALYZE		
:HARMONICS?	Queries all harmonic analysis settings.	5-64
:HARMONICS		
:CClass?	Queries all settings related to harmonic analysis class C.	5-64
:CClass		
:GETLambda	Sets the current power factor of harmonic analysis class C.	5-64
:LAMBda	Sets or queries the power factor of harmonic analysis class C.	5-64
:MAXCurrent	Sets or queries the fundamental current value of harmonic analysis class C.	5-64
:OPower	Sets or queries whether active power 25 W of harmonic analysis class C will be exceeded.	5-64
:CLASSs	Sets or queries the applicable class of the DUT in harmonic analysis.	5-64

5.1 List of Commands

Command	Function	Page
:DClass?	Queries all settings related to harmonic analysis class D.	5-64
:DClass		
:POWer	Sets or queries the power value of harmonic analysis class D.	5-65
:DMODE	Sets or queries the display mode of harmonic analysis.	5-65
:DETail?	Queries all settings related to the harmonic analysis result list.	5-65
:DETail		
:DISPlay	Sets or queries the display mode of the harmonic analysis result list.	5-65
:LIST		
:ITEM?	Queries all items that will be displayed in the harmonic analysis result list.	5-65
:VALue?	Queries all the data for the specified analysis number in the harmonic analysis result list.	5-65
:GROuping	Sets or queries the harmonic analysis grouping.	5-65
:IECEdition	Sets or queries measurement standard (IEC 61000-4-7) of the harmonic analysis measurement mode.	5-65
:RMS	Queries the rms value harmonic analysis.	5-66
:SPOint	Sets or queries the computation start point of harmonic analysis.	5-66
:SVOLtage	Sets or queries the supply voltage of harmonic analysis.	5-66
:THD	Queries the harmonic distortion factor of harmonic computation.	5-66
:I2T?	Queries all Joule integral settings.	5-66
:I2T		
:MATH	Sets or queries the Joule integral waveform display on/off state.	5-66
:MEASure?	Queries all settings related to the automated measurement of the Joule integral.	5-66
:MEASure		
:I2T?	Queries Joule integral settings.	5-66
:I2T		
:COUNT?	Queries the continuous statistical processing count of the Joule integral.	5-66
:{MAXimum MEAN MINimum SDEViation}?	Queries the Joule integral statistics.	5-66
:STATe	Sets or queries whether the Joule integral is to be measured.	5-66
:VALue?	Queries automatically measured value of the Joule integral.	5-67
:RANGe	Sets or queries the measurement source window.	5-67
:SCALe?	Queries all scaling settings.	5-67
:SCALe		
:CENTer	Sets or queries the center value for manual scaling.	5-67
:MODE	Sets or queries the scale mode.	5-67
:SENSitivity	Sets or queries the sensitivity for manual scaling.	5-67
:TRANge	Sets or queries the measurement time period.	5-67
:SETup?	Queries all power supply analysis input settings.	5-68
:SETup		
:ADESkew	Executes auto deskewing for power supply analysis.	5-68
:I?	Queries all current input channel settings for power supply analysis.	5-68
:I		
:DESKew	Sets or queries auto deskewing of the current input channel for power supply analysis.	5-68
:INPut	Sets or queries the current input channels for power supply analysis.	5-68
:PROBe	Sets or queries the current-to-voltage conversion ratio settings of the probe connected to the current input channel for power supply analysis.	5-68
:U?	Queries all voltage input channel settings for power supply analysis.	5-68
:U		
:DESKew	Sets or queries auto deskewing of the voltage input channel for power supply analysis.	5-68
:INPut	Sets or queries the voltage input channels for power supply analysis.	5-68
:PROBe	Sets or queries the probe attenuation of the voltage input channels for power supply analysis.	5-69
:RTRace	Sets or queries the deskewing source trace for power supply analysis.	5-69
:SOA?	Queries all settings related to the XY display for safe operating area analysis.	5-69
:SOA		
:CURSor?	Queries all settings related to the cursor measurement on the XY display for safe operating area analysis.	5-69
:CURSor		
:X<x>?	Queries all settings related to the horizontal cursors on the XY display for safe operating area analysis.	5-69

5.1 List of Commands

Command	Function	Page
:X<x>		
:POSition	Sets or queries the horizontal cursor positions on the XY display for safe operating area analysis.	5-69
:VALue?	Queries the horizontal cursor voltages on the XY display for safe operating area analysis.	5-69
:Y<x>?	Queries all settings related to the vertical cursors on the XY display for safe operating area analysis.	5-69
:Y<x>		
:POSition	Sets or queries the vertical cursor positions on the XY display for safe operating area analysis.	5-69
:VALue?	Queries the vertical cursor voltages on the XY display for safe operating area analysis.	5-70
:MODE	Sets or queries the automated measurement mode on the XY display for safe operating area analysis.	5-70
:TRANge	Sets or queries the VT waveform range to show on the XY display for safe operating area analysis.	5-70
:VTDisPlay	Sets or queries the on/off state of the VT waveform display on the XY display for safe operating area analysis.	5-70
:SWLoss?	Queries all switching loss settings.	5-70
:SWLoss		
:CCALc	Sets or queries the total loss equation.	5-70
:DPROximal?	Queries all distal, mesial, and proximal settings.	5-70
:DPROximal		
:MODE	Sets or queries the distal, mesial, and proximal point mode setting.	5-70
:PERCent	Sets or queries the distal, mesial, and proximal points as percentages.	5-71
:UNIT	Sets or queries the distal, mesial, and proximal points as voltages.	5-71
:ILEVel	Sets or queries the current level for determining the loss zero period of total loss.	5-71
:INDicator	Sets or queries the total loss measurement location indicator.	5-71
:MATH	Sets or queries the power waveform display on/off state.	5-71
:MEASure?	Queries all settings related to the automated measurement of power supply parameters.	5-71
:MEASure		
:<Parameter>?	Queries all power supply analysis parameters.	5-71
:<Parameter>		
:COUNT?	Queries the continuous statistical processing count of the power supply analysis parameter.	5-71
: {MAXimum MEAN MINimum SDEviation}?	Queries a statistical value of a power supply analysis parameter.	5-72
:STATe	Sets or queries the on/off state of the power supply analysis parameter.	5-72
:VALue?	Queries the automatically measured value of a power supply parameter.	5-72
:CONTinuous?	Queries all settings related to the normal statistical processing of automatically measured total loss.	5-72
:CONTinuous		
:REStArt	Restarts the normal statistical processing of automatically measured total loss.	5-72
:TLCHange	Sets or queries whether the normal statistical processing of automatically measured total loss is to be restarted.	5-73
:CYCLe		
:ABORt	Aborts the execution of the cyclic statistical processing of automatically measured total loss.	5-73
:EXECute	Executes the cyclic statistical processing of automatically measured total loss.	5-73
:HISTory		
:ABORt	Aborts the execution of the statistical processing of the history waveform of automatically measured total loss.	5-73
:EXECute	Executes the statistical processing of the history waveform of automatically measured total loss.	5-73
:MODE	Sets or queries the statistical processing type of automatically measured total loss.	5-73
:ZLINKage	Sets or queries the linkage (on/off) between the result numbers and zoom positions of the results of automatically measured total loss.	5-73
:METHod	Sets or queries the calculation method for high and low points.	5-73
:RANGe	Sets or queries the measurement source window.	5-73

5.1 List of Commands

Command	Function	Page
:RDS	Sets or queries the total loss on-resistance.	5-74
:SCALe?	Queries all scaling settings.	5-74
:SCALe		
:CENTer	Sets or queries the center value for manual scaling.	5-74
:MODE	Sets or queries the scale mode.	5-74
:SENSitivity	Sets or queries the sensitivity for manual scaling.	5-74
:TRANge	Sets or queries the measurement time period.	5-74
:ULEVel	Sets or queries the voltage level for determining the loss calculation period of total loss.	5-74
:UNIT	Sets or queries the watt-hour unit.	5-74
:VCE	Sets or queries the collector-emitter saturation voltage of total loss.	5-75
:TYPE	Sets or queries the power supply analysis type.	5-75
:PMEASURE<x>?	Queries all power measurement settings.	5-75
:PMEASURE<x>		
:CYCMode	Sets or queries power measurement cycle mode.	5-75
:CYCTrace	Sets or queries the source waveform of power measurement cycle mode.	5-75
:IDPROximal?	Queries all distal, mesial, and proximal settings.	5-75
:IDPROximal		
:MODE	Sets or queries the distal, mesial, and proximal point mode setting.	5-75
:PERCent	Sets or queries the distal, mesial, and proximal points as percentages.	5-75
:UNIT	Sets or queries the distal, mesial, and proximal points as voltages.	5-75
:IMETHod	Sets or queries the high and low points.	5-76
:INDicator	Sets or queries the measurement location indicator.	5-76
:MEASURE?	Queries all settings related to the automated measurement of power measurement parameters.	5-76
:MEASURE		
:<Parameter>?	Queries the power measurement parameter settings.	5-76
:<Parameter>		
:COUNT?	Queries the normal statistical processing count of the power measurement parameter.	5-76
:{MAXimum MEAN MINimum SDEVIation}?	Queries the statistical value of a power measurement parameter.	5-76
:STATE	Sets or queries the on/off state of a power measurement parameter.	5-76
:VALue?	Queries the automatically measured power measurement parameter value.	5-77
:ALL	Collectively turns on or off the power measurement parameters.	5-77
:CONTinuous	Queries all settings related to the normal statistical processing of automated power measurement.	5-77
:REStart	Restarts the normal statistical processing of automated power measurement.	5-77
:TLCHange	Sets or queries whether the normal statistical processing of automated power measurement is to be restarted.	5-77
:CYCLe?	Queries all settings related to the cyclic statistical processing of automated power measurement.	5-77
:CYCLe		
:ABORT	Aborts the execution of the cyclic statistical processing of automated power measurement.	5-77
:EXECute	Executes the cyclic statistical processing of automated power measurement.	5-77
:TRACe	Sets or queries the source waveform of the cyclic statistical processing of automated power measurement.	5-78
:HISTory		
:ABORT	Aborts the statistical processing of the history waveform of automated power measurement.	5-78
:EXECute	Executes the statistical processing of the history waveform of automated power measurement.	5-78
:MODE	Sets or queries the statistical processing type of automated power measurement.	5-78
:ZLINKage	Sets or queries the linkage (on/off) between the result numbers and zoom positions of the results of automated power measurement.	5-78
:MODE	Sets or queries the on/off state of power measurement.	5-78
:RANGe	Sets or queries the measurement source window.	5-78
:SETUp?	Queries all power measurement input settings.	5-78
:SETUp		
:ADESkew	Executes auto deskewing for power measurement.	5-78
:I?	Queries all current input channel settings for power measurement.	5-79

5.1 List of Commands

Command	Function	Page
:I		
:DESKew	Sets or queries deskewing of the current input channel for power measurement.	5-79
:PROBe	Sets or queries the current-to-voltage conversion ratio settings of the probe connected to the current input channel for power measurement.	5-79
:RTRace	Sets or queries the deskewing source trace for power measurement.	5-79
:U?	Queries all voltage input channel settings for voltage measurement.	5-79
:U		
:DESKew	Sets or queries auto deskewing of the voltage input channel for power measurement.	5-79
:PROBe	Sets or queries the probe attenuation of the voltage input channels for power measurement.	5-79
:TRANge	Sets or queries the measurement time period.	5-79
:UDPRoximal?	Queries all distal, mesial, and proximal settings.	5-79
:UDPRoximal		
:MODE	Sets or queries the distal, mesial, and proximal point mode setting.	5-80
:PERCent	Sets or queries the distal, mesial, and proximal points as percentages.	5-80
:UNIT	Sets or queries the distal, mesial, and proximal points as voltages.	5-80
:UMETHod	Sets or queries the high and low points.	5-80
:UNIT	Sets or queries the watt-hour unit.	5-80
:USER<y>?	Queries all automatic measurement settings for a Calc item.	5-80
:USER<y>		
:COUNT?	Queries the statistical processing count of the automatically measured value of a Calc item.	5-80
:DEFine	Sets or queries the expression for the automatically measured value of a Calc item.	5-80
:{MAXimum MEAN MINimum S DEVIation}?	Queries a statistical value that is calculated on the automatically measured value of a Calc item.	5-81
:NAME	Sets or queries the name of a Calc item.	5-81
:STATe	Sets or queries the on/off status of automated measurement of a Calc item.	5-81
:UNIT	Sets or queries the unit of a Calc item.	5-81
:VALue?	Queries the automatically measured value of a Calc item.	5-81
:WAIT?	Waits for the completion of automated measurement with a timeout.	5-81

ASETup Group

:ASETup		
:EXECute	Executes auto setup.	5-82
:UNDO	Undoes auto setup.	5-82

CALibrate Group

:CALibrate?	Queries all calibration settings.	5-83
:CALibrate		
[:EXECute]	Executes calibration.	5-83
:MODE	Sets or queries the auto calibration on/off state.	5-83

CHANnel Group

:CHANnel<x>?	Queries all vertical axis settings of a channel.	5-84
:CHANnel<x>		
:AScale		
[:EXECute]	Executes auto scaling of a channel.	5-84
:BWIDTH	Sets or queries the input filter of the specified channel.	5-84
:COUPling	Sets or queries the input coupling of the specified channel.	5-84
:DESKew	Sets or queries the deskewing of the specified channel.	5-84
:DISPlay	Sets or queries the display on/off state of the specified channel.	5-84
:INVert	Sets or queries the on/off state of the invert mode (waveform inversion display).	5-84
:LABel?	Queries all settings related to the waveform label of the specified channel.	5-84
:LABel		
:DEFine	Sets or queries the waveform label of a channel.	5-84
:DISPlay	Sets or queries the display on/off state of the specified channel's waveform label.	5-85
:LSCale?	Queries all the linear scaling settings of the specified channel.	5-85

5.1 List of Commands

Command	Function	Page
:LSCale		
:AVALue	Sets or queries the scaling coefficient A.	5-85
:BVALue	Sets or queries offset B.	5-85
:MODE	Sets or queries the on/off state of linear scaling.	5-85
:UNIT	Sets or queries a unit that is attached to linear scaling results.	5-85
:OFFSet	Sets or queries the offset voltage of the specified channel.	5-85
:POSition	Sets or queries the vertical position of the specified channel.	5-85
:PROBe?	Queries all settings related to the probe attenuation of the specified channel.	5-85
:PROBe		
:DZCalibrate	Executes demagnetization and zero calibration of the specified channel.	5-85
[:MODE]	Sets or queries the probe attenuation of the specified channel.	5-86
:PZCalibrate	Executes zero calibration of the specified channel's current probe.	5-86
:VARIABLE	Sets or queries the specified channel's voltage scale in 0.01 V/div steps.	5-86
:VDIV	Sets or queries the specified channel's voltage scale (V/div).	5-86

CLEAr Group

:CLEAr	Clears traces.	5-87

COMMunicate Group

:COMMunicate?	Queries all communication settings.	5-88
:COMMunicate		
:HEADer	Sets or queries whether a header is added to the response to a query (example with header: "CHANNEL1:PROBE:MODE 10"; example without header: "10").	5-88
:LOCKout	Sets/clears local lockout.	5-88
:OPSE	Sets or queries the overlap command that is used by the *OPC, *OPC?, and *WAI commands.	5-88
:OPSR?	Queries the operation pending status register.	5-88
:OVERlap	Sets or queries the commands that operate as overlap commands.	5-88
:REMOte	Sets the instrument to remote or local mode. On is remote mode.	5-88
:VERBose	Sets or queries whether the response to a query is returned fully spelled out (e.g., CHANNEL1:PROBE:MODE 10) or in its abbreviated form (e.g., CHAN:PROB 10).	5-89
:WAIT	Waits for a specified extended event to occur.	5-89
:WAIT?	Creates the response that is returned when a specified extended event occurs.	5-89

CURSor Group

:CURSor?	Queries all cursor measurement settings.	5-90
:CURSor		
[:TY]?	Queries all cursor settings.	5-90
[:TY]		
:DEGRee?	Queries all angle cursor settings.	5-90
:DEGRee		
:ALL	Collectively turns on or off the measured values of the angle cursor on the VT display.	5-90
:D<x>?	Queries all angle cursor's angle settings.	5-90
:D<x>		
:STATe	Sets or queries the on/off state of the angle cursor's angle.	5-90
:VALue?	Queries the angle cursor's angle.	5-90
:DD?	Queries all settings related to the angle difference between the angle cursors.	5-90
:DD		
:STATe	Sets or queries the on/off state of the angle difference ΔD value between the angle cursors.	5-90
:VALue?	Queries the angle difference ΔD value between the angle cursors.	5-90
:DV?	Queries all settings related to ΔV between the angle cursors.	5-90
:DV		
:STATe	Sets or queries the on/off state of the ΔV value between the angle cursors.	5-90
:VALue?	Queries the ΔV value between the angle cursors.	5-91
:JUMP	Moves the angle cursor on the VT display on to the specified zoom waveform.	5-91

5.1 List of Commands

Command	Function	Page
:POSition<x>	Sets or queries the angle cursor position.	5-91
:REFerence<x>	Sets or queries the start point (Reference1) or end point (Reference2) of the reference angle.	5-91
:RVALue	Sets or queries the reference angle.	5-91
:TRACe	Sets or queries the source waveform of the angle cursor.	5-91
:UNIT	Sets or queries the angle cursor unit.	5-91
:V<x>?	Queries all angle cursor's voltage settings.	5-91
:V<x>		
:STATe	Sets or queries the on/off state of the angle cursor's voltage.	5-92
:VALue?	Queries the angle cursor's voltage.	5-92
:HORizontal?	Queries all ΔV cursor settings.	5-92
:HORizontal		
:ALL	Collectively turns on or off the measured values of the ΔV cursor on the VT display.	5-92
:DV?	Queries all settings related to the vertical value between the ΔV cursors.	5-92
:DV		
:STATe	Sets or queries the on/off state of the vertical value between the ΔV cursors.	5-92
:VALue?	Queries the vertical value between the ΔV cursors.	5-92
:DVDT?	Queries all settings related to the ΔV cursor's $\Delta V/\Delta T$ value.	5-92
:DVDT		
:STATe	Sets or queries the on/off state of the ΔV cursor's $\Delta V/\Delta T$ value.	5-92
:VALue?	Queries the ΔV cursor's $\Delta V/\Delta T$ value.	5-92
:POSition<x>	Sets or queries the ΔV cursor position.	5-92
:TRACe	Sets or queries the source waveform of the ΔV cursor.	5-93
:V<x>?	Queries all settings related to the ΔV cursor's vertical value.	5-93
:V<x>		
:STATe	Sets or queries the on/off state of the ΔV cursor's vertical value.	5-93
:VALue?	Queries the ΔV cursor's vertical value.	5-93
:MARKer?	Queries all marker cursor settings.	5-93
:MARKer		
:FORM	Sets or queries the display format of the marker cursor.	5-93
:M<x>?	Queries all settings related to the specified marker cursor.	5-93
:M<x>		
:ALL	Collectively turns on or off the measured values of the marker cursor on the VT display.	5-93
:DT<y>?	Queries all settings related to the time value between the marker cursors.	5-93
:DT<y>		
:STATe	Sets or queries the on/off state of the time value between the marker cursors.	5-93
:VALue?	Queries the time value between the marker cursors.	5-93
:DV<y>?	Queries all settings related to the vertical value between the marker cursors.	5-93
:DV<y>		
:STATe	Sets or queries the on/off state of the vertical value between the marker cursors.	5-94
:VALue?	Queries the vertical value between the marker cursors.	5-94
:JUMP	Moves the marker cursor on the VT display on to the specified zoom waveform.	5-94
:POSition	Sets or queries the marker cursor's time value.	5-94
:T?	Queries all settings related to the marker cursor's time value.	5-94
:T		
:STATe	Sets or queries the on/off state of the marker cursor's time value.	5-94
:VALue?	Queries the marker cursor's time value.	5-94
:TRACe	Sets or queries the source waveform of the marker cursor.	5-94
:V?	Queries all settings related to the marker cursor's vertical value.	5-94
:V		
:STATe	Sets or queries the on/off state of the marker cursor's vertical value.	5-94
:VALue?	Queries the marker cursor's vertical value.	5-95
:TYPE	Sets or queries the cursor type.	5-95
:VERTical?	Queries all ΔT cursor settings.	5-95
:VERTical		
:ALL	Collectively turns on or off the measured values of the ΔT cursor on the VT display.	5-95
:DT?	Queries all settings related to the time value between the ΔT cursors.	5-95

5.1 List of Commands

Command	Function	Page
:DT		
:STATe	Sets or queries the on/off state of the time value between the ΔT cursors.	5-95
:VALue?	Queries the time value between the ΔT cursors.	5-95
:DV?	Queries all settings related to the vertical value between the ΔT cursors.	5-95
:DV		
:STATe	Sets or queries the on/off state of the vertical value between the ΔT cursors.	5-95
:VALue?	Queries the vertical value between the ΔT cursors.	5-95
:DVDT?	Queries all settings related to the ΔT cursor's $\Delta V/\Delta T$ value.	5-95
:DVDT		
:STATe	Sets or queries the on/off state of the ΔT cursor's $\Delta V/\Delta T$ value.	5-95
:VALue?	Queries the ΔT cursor's $\Delta V/\Delta T$ value.	5-96
:JUMP	Moves the vertical cursor on the VT display on to the specified zoom waveform.	5-96
:PERDt?	Queries all settings related to the $1/\Delta T$ value between the ΔT cursors.	5-96
:PERDt		
:STATe	Sets or queries the on/off state of the $1/\Delta T$ value between the ΔT cursors.	5-96
:VALue?	Queries the $1/\Delta T$ value between the ΔT cursors.	5-96
:POSition<x>	Sets or queries the ΔT cursor position.	5-96
:T<x>?	Queries all settings related to the ΔT cursor's time value.	5-96
:T<x>		
:STATe	Sets or queries the on/off state of the ΔT cursor's time value.	5-96
:VALue?	Queries the ΔT cursor's time value.	5-96
:TRACe	Sets or queries the source waveform of the ΔT cursor.	5-96
:V<x>?	Queries all settings related to the ΔT cursor's vertical value.	5-96
:V<x>		
:STATe	Sets or queries the on/off state of the ΔT cursor's vertical value.	5-96
:VALue?	Queries the ΔT cursor's vertical value.	5-97

DISPlay Group

:DISPlay?	Queries all display settings.	5-98
:DISPlay		
:ACCumulate?	Queries all accumulated waveform display settings.	5-98
:ACCumulate		
:MODE	Sets or queries the accumulation mode.	5-98
:PERSistence	Sets or queries the accumulation time.	5-98
:COLor?	Queries all waveform color settings.	5-98
:COLor		
:{CHANnel<x> MATH<x>}	Sets or queries a waveform color.	5-98
:LState	Sets or queries the logic waveform state display color.	5-98
:SERialbus<x>?	Queries the color settings of all trends of the specified serial bus.	5-98
:SERialbus<x>		
:TREnd<y>	Sets or queries the serial bus trend color.	5-98
:FGRid	Sets or queries whether the fine grid is displayed.	5-98
:FORMat	Sets or queries the display format.	5-99
:GRATicule	Sets or queries the graticule (grid).	5-99
:INTENsity?	Queries all settings related to the intensity of display items.	5-99
:INTENsity		
[:WAVeform]	Sets or queries the waveform intensity.	5-99
:{CURSor GRID MARKer ZBOX}	Sets or queries the intensity of a display item.	5-99
:INTERpolate	Sets or queries the interpolation method.	5-99
:MAPPING	Sets or queries the split screen waveform mapping mode.	5-99
:SMAPPING?	Queries all waveform mappings to the split screen.	5-99
:SMAPPING		
:{CHANnel<x> MATH<x>}	Sets or queries the specified waveform mapping to the split screen.	5-99
:SVALue	Sets or queries the on/off status of the scale value display.	5-99

FFT Group

:FFT<x>?	Queries all FFT analysis settings.	5-100
:FFT<x>		
:AVERage?	Queries all settings related to the averaging of FFT analysis.	5-100
:AVERage		

5.1 List of Commands

Command	Function	Page
:EWEight	Sets or queries the attenuation constant of exponential averaging of FFT analysis.	5-100
:DATA?	Queries all settings related to the FFT waveform data that will be sent.	5-100
:DATA		
:BYTeorder	Sets or queries the transmission byte order when the FFT waveform data format is binary.	5-100
:END	Sets or queries the end point of the FFT waveform data that will be sent.	5-100
:FORMat	Sets or queries the format of the FFT waveform data that will be sent.	5-100
:LENGth?	Queries the total number of data points of the FFT waveform that will be sent.	5-100
:SEND?	Queries the FFT waveform data.	5-100
:START	Sets or queries the start point of the FFT waveform data that will be sent.	5-101
:DISPlay	Sets or queries whether FFT analysis is to be performed.	5-101
:HORizontal?	Queries all horizontal axis settings of FFT analysis.	5-101
:HORizontal		
:CSPan?	Queries all settings related to the center point and span of the horizontal axis in FFT analysis.	5-101
:CSPan		
:CENTer	Sets or queries the center value of the horizontal axis in FFT analysis.	5-101
:SPAN	Sets or queries the span of the horizontal axis in FFT analysis.	5-101
:LRIGht?	Queries all settings related to the left and right edges of the horizontal axis in FFT analysis.	5-101
:LRIGht		
:RANGe	Sets or queries the range of the left and right edges of the horizontal axis in FFT analysis.	5-101
:MODE	Sets or queries the horizontal axis mode of FFT analysis.	5-101
:LENGth	Sets or queries the number of FFT points of FFT analysis.	5-101
:MEASure?	Queries all automated measurement settings of FFT analysis.	5-102
:MEASure		
:MARKer?	Queries all settings related to the marker cursor measurement of FFT analysis.	5-102
:MARKer		
[:BASic]?	Queries all settings related to the basic marker cursor items of FFT analysis.	5-102
[:BASic]		
:ALL	Collectively turns on or off all the basic marker cursor items of FFT analysis.	5-102
:DFRequency?	Queries all settings related to the frequency between marker cursors of FFT analysis.	5-102
:DFRequency		
:STATe	Sets or queries the on/off state of the frequency between marker cursors of FFT analysis.	5-102
:VALue?	Queries the frequency between marker cursors of FFT analysis.	5-102
:DV?	Queries all settings related to the level between marker cursors of FFT analysis.	5-102
:DV		
:STATe	Sets or queries the on/off state of the level between marker cursors of FFT analysis.	5-102
:VALue?	Queries the level between marker cursors of FFT analysis.	5-102
:FREQuency<y>?	Queries all settings related to the specified marker cursor's frequency of FFT analysis.	5-102
:FREQuency<y>		
:STATe	Sets or queries the on/off state of the specified marker cursor's frequency of FFT analysis.	5-103
:VALue?	Queries the specified marker cursor's frequency of FFT analysis.	5-103
:POSition<y>	Sets or queries the specified marker cursor's position of FFT analysis.	5-103
:V<y>?	Queries all settings related to the specified marker cursor's level of FFT analysis.	5-103
:V<y>		
:STATe	Sets or queries the on/off state of the specified marker cursor's level of FFT analysis.	5-103
:VALue?	Queries the specified marker cursor's level of FFT analysis.	5-103
:MODE	Sets or queries the automated measurement mode of FFT analysis.	5-103
:PARAmeter?	Queries all settings related to the FFT analysis parameters.	5-103
:PARAmeter		

5.1 List of Commands

Command	Function	Page
:OVERall?	Queries all settings related to the overall value, an FFT analysis parameter.	5-103
:OVERall		
:STATe	Sets or queries the on/off state of the overall value, an FFT analysis parameter.	5-104
:VALue?	Queries the overall value, an FFT analysis parameter.	5-104
:PEAK?	Queries all settings related to the peak value measurement of FFT analysis.	5-104
:PEAK		
:DETail?	Queries all FFT analysis result settings.	5-104
:DETail		
:LIST?	Queries all settings related to the FFT analysis result list.	5-104
:LIST		
:ITEM?	Queries all items that will be displayed in the FFT analysis result list.	5-104
:MAXPeak		
:EXECute	Moves to the maximum peak value in the FFT analysis results.	5-104
:NUMBer	Sets or queries the analysis number in the FFT analysis result list.	5-104
:VALue?	Queries all the data for the specified analysis number in the FFT analysis result list.	5-104
:EXCursion	Sets or queries the difference between peak and valley of the peak value of FFT analysis.	5-105
:FVALue?	Queries the peak frequency of FFT analysis.	5-105
:RANGe	Sets or queries the measurement frequency range of the peak value of FFT analysis.	5-105
:THReshold	Sets or queries the threshold level of the peak value of FFT analysis.	5-105
:VVALue?	Queries the peak value of FFT analysis.	5-105
:MODE	Sets or queries the waveform display mode of FFT analysis.	5-105
:RANGe	Sets or queries the measurement source window of FFT analysis.	5-105
:RPOSition	Sets or queries the center zoom point on the vertical axis in FFT analysis.	5-105
:TRACe	Sets or queries the source waveform of FFT analysis.	5-105
:TYPE	Sets or queries the spectrum of FFT analysis.	5-106
:UNIT?	Queries all FFT computation unit settings.	5-106
:UNIT		
:DBM?	Queries all settings for when the FFT computation unit is dBm.	5-106
:DBM		
:IMPedance	Sets or queries the impedance when the FFT computation unit is dBm.	5-106
:MODE	Sets or queries whether an FFT computation unit will be attached automatically or manually.	5-106
:VERTical?	Queries all vertical axis settings of FFT analysis.	5-106
:VERTical		
:LEVel	Sets or queries the vertical axis display position of FFT analysis.	5-106
:MODE	Sets or queries the vertical axis mode of FFT analysis.	5-106
:SENSitivity	Sets or queries the vertical axis sensitivity of FFT analysis.	5-106
:VTDisplay	Sets or queries the on/off state of the VT waveform display of FFT analysis.	5-107
:WINDow	Sets or queries the window function of FFT analysis.	5-107

FILE Group

:FILE?	Queries all data storage settings.	5-108
:FILE		
:COPY		
:ABORt	Aborts file copying.	5-108
:CDIRectory	Changes the file copy destination directory.	5-108
:DRIVe	Sets the storage device to copy files to.	5-108
[[:EXECute]	Executes file copying. This is an overlap command.	5-108
:PATH?	Queries the file copy destination directory.	5-108
:DELeTe		
{:AHIStogram ASCIi BINary BMP FFT HLISt JPEg MEASure PNG SBUS SETup SNAP ZPOLygon ZWAVE}		
[[:EXECute]	Deletes a specific type of file. This is an overlap command.	5-108
[[:DIRectory]		
:CDIRectory	Changes the current directory.	5-108
:DRIVe	Sets the current storage device.	5-108

5.1 List of Commands

Command	Function	Page
:FREE?	Queries the free space on the current storage device that is being operated on in bytes.	5-108
:MDIRectory	Creates a directory in the current location. This is an overlap command.	5-108
:PATH?	Queries the current directory.	5-109
:LOAD		
:BINary		
:ABORt	Aborts the loading of waveform data.	5-109
[:EXECute]	Executes the loading of waveform data. This is an overlap command.	5-109
: {SETup SNAP ZPOLygon<x> ZWAVE<x>}		
:ABORt	Aborts the loading of a specific type of file.	5-109
[:EXECute]	Executes the loading of a specific type of file. This is an overlap command.	5-109
:MOVE		
:ABORt	Aborts file moving.	5-109
:CDIRectory	Changes the file move destination directory.	5-109
:DRIVe	Sets the storage device to move files to.	5-109
[:EXECute]	Executes file moving. This is an overlap command.	5-109
:PATH?	Queries the file move destination directory.	5-109
:PROTect		
[:EXECute]	Turns on or off the file protection.	5-109
:REName	Renames the file.	5-109
:SAVE?	Queries all file save settings.	5-109
:SAVE		
: {AHIStogram ASCIi BINary FFT HARMonics HLISt SBUS ZWAve}?	Queries all the settings related to the saving of a specific type of file.	5-109
: {AHIStogram ASCIi BINary FFT HARMonics HLISt MEASure SBUS SETup SNAP ZWAVE}		
:ABORt	Aborts the saving of a specific type of file.	5-110
: {AHIStogram ASCIi BINary FFT HARMonics HLISt MEASure SBUS SETup SNAP ZWAVE}		
[:EXECute]	Executes the saving of a specific type of file. This is an overlap command.	5-110
: {AHIStogram FFT HARMonics SBUS ZWAVE}		
:SElect	Sets or queries the area for saving a specific type of file.	5-110
:ANAMing	Sets or queries the on/off state of the auto generation of the name of the file to be saved.	5-110
: {ASCIi BINary}		
:COMPreSSion	Sets or queries the save compression method for a specific type of file.	5-110
: {ASCIi BINary SBUS}		
:HISToRy	Sets or queries the save method for the specified history waveform.	5-110
: {ASCIi BINary}		
:LENGth	Sets or queries the number of data points to save when compressing or sampling (decimating) various types of data.	5-110
:RANGe	Sets or queries the window for saving a specific type of data.	5-110
:ASCIi		
:TINformation	Sets or queries whether waveform data is saved with time information (ON) or without it (OFF).	5-111
: {ASCIi BINary}		
:TRACe	Sets or queries the waveform for saving a specific type of data.	5-111
:COMMeNt	Sets or queries the comment for the data to be saved.	5-111
:FFT		
:FINformation	Sets or queries whether FFT data is saved with frequency information (ON) or without it (OFF).	5-111
:NAME	Sets or queries the file name for the data to be saved.	5-111
GONogo Group		
:GONogo?	Queries all GO/NO-GO determination settings.	5-112
:GONogo		
:ABORt	Aborts GO/NO-GO determination.	5-112

5.1 List of Commands

Command	Function	Page
:ACTion?	Queries all settings for the action that is performed when the judgment is NO-GO and the reference for the judgment.	5-112
:ACTion		
:BUZZer	Sets or queries whether a warning sound is to be generated when the judgment is NO-GO.	5-112
:HCOPy	Sets or queries whether a hard copy is output to the built-in printer (option) or an external printer when the judgment is NO-GO.	5-112
:MAIL?	Queries all settings related to the mail notification that is made when the judgment is NO-GO.	5-112
:MAIL		
:COUNT	Sets or queries the upper limit to the number of mail notifications made when the judgment is NO-GO.	5-112
:MODE	Sets or queries whether a mail notification is to be made when the judgment is NO-GO.	5-112
:SAVE	Sets or queries whether the event is to be saved to the storage device when the judgment is NO-GO.	5-112
:COUNT?	Queries the number of GO/NO-GO determinations that were performed.	5-112
:EXECute	Executes GO/NO-GO determination. This is an overlap command.	5-112
:LOGic	Sets or queries the GO/NO-GO determination type.	5-112
:NGCount?	Queries the GO/NO-GO determination NO-GO count.	5-112
:NGStopcount	Sets or queries the number of NO-GO results.	5-113
:STOPcount	Sets or queries the number of determination results acquired.	5-113
:WAIT?	Waits for the completion of GO/NO-GO determination with a timeout.	5-113
[:ZPARAMeter] ?	Queries all zone or parameter determination settings.	5-113
[:ZPARAMeter]		
:NUMBer<x>?	Queries all settings related to the zone or parameter determination conditions.	5-113
:NUMBer<x>		
:CAUSE?	Queries whether the specified waveform parameter for zone or parameter determination is the cause of a NO-GO judgment.	5-113
:CONDition	Sets or queries the specified waveform parameter's reference condition for zone or parameter determination.	5-113
:MODE	Sets or queries the mode of the specified condition.	5-113
:PARAMeter?	Queries all the parameter settings of the specified condition.	5-113
:PARAMeter		
:ITEM	Sets or queries the specified waveform parameter's item for parameter determination.	5-114
:LIMit	Sets or queries the specified waveform parameter's upper and lower limits for parameter determination.	5-114
:TRACe	Sets or queries the specified waveform parameter's source waveform for parameter determination.	5-114
:VALue?	Queries the measured value of the specified waveform parameter for parameter determination.	5-114
:POLYgon?	Queries all polygonal zone determination settings.	5-114
:POLYgon		
:HPOSition	Sets or queries the horizontal position used in polygonal zone determination.	5-115
:RANGe	Sets or queries the source window used in polygonal zone determination.	5-115
:TRACe	Sets or queries the source waveform used in polygonal zone determination.	5-115
:VPOSition	Sets or queries the vertical position used in polygonal zone determination.	5-115
:ZNUMber	Sets or queries the zone number used in polygonal zone determination.	5-115
:RECTangle?	Queries all waveform rectangular zone determination settings.	5-115
:RECTangle		
:HORizontal	Sets or queries the horizontal position of the rectangle used in rectangular zone determination.	5-116
:RANGe	Sets or queries the source window of the rectangle used in rectangular zone determination.	5-116
:TRACe	Sets or queries the source waveform of the rectangle used in rectangular zone determination.	5-116
:VERTical	Sets or queries the vertical position of the rectangle used in rectangular zone determination.	5-116
:WAVE?	Queries all waveform zone determination settings.	5-116
:WAVE		
:EDIT		

5.1 List of Commands

Command	Function	Page
:EXIT	Exits from the waveform zone edit menu.	5-116
:NEW	Sets the base waveform in waveform zone editing.	5-116
:PART	Executes partial editing in waveform zone editing.	5-117
:WHOLe	Sets the entire waveform zone in waveform zone editing.	5-117
:RANGe	Sets or queries the source window used in waveform zone determination.	5-117
:TRACe	Sets or queries the source waveform used in waveform zone determination.	5-117
:TRANge	Sets or queries the determination period used in waveform zone determination.	5-117
:ZNUMber	Sets or queries the zone number used in waveform zone determination.	5-117

HCOPy group

:HCOPy?	Queries all screen capture data output settings.	5-118
:HCOPy		
:ABORt	Stops data output and paper feeding.	5-118
:DIRectiOn	Sets or queries the data output destination.	5-118
:EXECute	Executes data output.	5-118
:EXTPrinter?	Queries all external printer output settings.	5-118
:EXTPrinter		
:MODE	Sets or queries the normal or hard copy setting for outputting to the external printer.	5-118
:TONE	Sets or queries the external printer output color.	5-118
:MULTitarget?	Queries all multi target feature settings.	5-118
:MULTitarget		
:EXTPrinter	Sets or queries whether the multi target feature will send output to the external printer.	5-118
:PRINter	Sets or queries whether the multi target feature will send output to the built-in printer.	5-118
:NETPrinter	Sets or queries whether the multi target feature will send output to a network printer.	5-118
:FILE	Sets or queries whether the multi target feature will send output to a file.	5-118
:WAVEform	Sets or queries whether the multi target feature will send output to a waveform file.	5-119
:NETPrint?	Queries all network printer output settings.	5-119
:NETPrint		
:MODE	Sets or queries the normal or hard copy setting for outputting to the network printer.	5-119
:TONE	Sets or queries the network printer output color.	5-119
:TYPE	Sets or queries the network printer output command type.	5-119
:PRINter?	Queries all built-in printer output settings.	5-119
:PRINter		
:MODE	Sets or queries the short or hard copy setting for outputting to the built-in printer.	5-119
:REPort	Sets or queries whether additional information is to be output to the built-in printer.	5-119

HISTory Group

:HISTory?	Queries all history feature settings.	5-120
:HISTory		
:AVERage	Sets or queries the history waveform highlight display mode.	5-120
:DISPlay	Sets or queries the history record start and end numbers that will be displayed.	5-120
:DMODE	Sets or queries the history waveform display mode.	5-120
:RECOrd	Sets or queries the history waveform source record.	5-120
:REPLay?	Queries all history waveform replay settings.	5-120
:REPLay		
:JUMP	Moves to the specified record number in the history waveform.	5-120
:SPEEd	Sets or queries the history waveform replay speed.	5-120
:STARt	Starts replaying the history waveform in the specified direction.	5-120
:STOP	Stops replaying the history waveform.	5-120
:SEARCh?	Queries all history waveform search settings.	5-120
[:SEARCh]		
:ABORt	Aborts the search.	5-120

5.1 List of Commands

Command	Function	Page
:EXECute	Executes a search. This is an overlap command.	5-121
:LOGic	Sets or queries the history waveform search logic.	5-121
:NUMBer<x>?	Queries all search condition settings.	5-121
:NUMBer<x>		
:CONDition	Sets or queries the reference condition of the specified search condition.	5-121
:MODE	Sets or queries the mode of the specified search condition.	5-121
:PARAmeter?	Queries all parameter search settings.	5-121
:PARAmeter		
:ITEM	Sets or queries the specified waveform parameter's item for parameter searching.	5-121
:LIMit	Sets or queries the specified waveform parameter's upper and lower limits for parameter searching.	5-121
:TRACe	Sets or queries the specified waveform parameter's source waveform for parameter searching.	5-122
:VALue?	Queries the measured value of the specified waveform parameter for parameter searching.	5-122
:POLYgon?	Queries all polygonal zone search settings.	5-122
:POLYgon		
:HPOSition	Sets or queries the horizontal position used in polygonal zone searching.	5-122
:RANGe	Sets or queries the source window used in polygonal zone searching.	5-122
:TRACe	Sets or queries the source waveform used in polygonal zone searching.	5-122
:VPOSition	Sets or queries the vertical position used in polygonal zone searching.	5-123
:ZNUMber	Sets or queries the zone number used in polygonal zone searching.	5-123
:RECTangle?	Queries all waveform rectangular zone search settings.	5-123
:RECTangle		
:HORizontal	Sets or queries the horizontal position of the rectangle used in rectangular zone searching.	5-123
:RANGe	Sets or queries the source window of the rectangle used in rectangular zone searching.	5-123
:TRACe	Sets or queries the source waveform of the rectangle used in rectangular zone searching.	5-123
:VERTical	Sets or queries the vertical position of the rectangle used in rectangular zone searching.	5-124
:WAVE?	Queries all waveform zone search settings.	5-124
:WAVE		
:RANGe	Sets or queries the source window used in waveform zone searching.	5-124
:TRACe	Sets or queries the source waveform used in waveform zone searching.	5-124
:TRANge	Sets or queries the determination period used in waveform zone searching.	5-124
:ZNUMber	Sets or queries the zone number used in waveform zone searching.	5-124
:RESet	Resets the history waveform search conditions.	5-124
:SIMPlE?	Queries all simple searching settings.	5-124
:SIMPlE		
:HORizontal	Sets or queries the horizontal position of the rectangle to use in simple searching.	5-124
:RANGe	Sets or queries the target window of the rectangle to use in simple searching.	5-125
:TRACe	Sets or queries the source trace of the rectangle to use in simple searching.	5-125
:VERTical	Sets or queries the vertical position of the rectangle to use in simple searching.	5-125
:TIME?	Queries the time of the source record number.	5-125

IMAGe Group

:IMAGe?	Queries all screen capture data output settings.	5-126
:IMAGe		
:ABORt	Aborts saving screen capture data to the storage device.	5-126
:BACKground	Sets or queries the screen capture background.	5-126
:EXECute	Saves the screen capture data to the storage device.	5-126
:FORMat	Sets or queries the screen capture output format.	5-126
:INFormation	Sets or queries whether setting information is included in screen capture data.	5-126
:MODE	Sets or queries the screen capture output mode.	5-126
:SAVE?	Queries all file output settings.	5-126
:SAVE		

5.1 List of Commands

Command	Function	Page
:ANAMing	Sets or queries the on/off state of the auto naming feature for saving files.	5-126
:CDIRectory	Changes the file directory.	5-126
:DRIVE	Sets the storage device that the file will be created in.	5-126
:NAME	Sets or queries the name of the file to be created.	5-126
:SEND?	Queries the screen capture data value.	5-127
:TONE	Sets or queries the color tone of the screen capture data that will be saved.	5-127
:WGRadation	Sets or queries whether or not gradation is applied to the waveform portion of the screen image data to be output.	5-127

INITialize Group

:INITialize		
:EXECute	Executes initialization.	5-128
:UNDO	Undoes the initialization that was executed.	5-128

LOGic group

:LOGic?	Queries all logic input waveform settings.	5-129
:LOGic		
:MODE	Sets or queries the logic input on/off state.	5-129
:PODA?	Queries all logic input settings.	5-129
[:PODA]		
:BITOrder	Sets or queries bit order of the logic input.	5-129
:BITSetup		
:ALL		
:DISPlay	Turns on or off all bit displays of the specified pod of the logic input.	5-129
:LEVel	Sets or queries the user-defined threshold level of the specified pod of the logic input.	5-129
:TYPE	Selects the threshold level of the specified pod of the logic input.	5-129
:BIT<x>?	Queries all settings of a bit of the specified pod of the logic input.	5-129
:BIT<x>		
:DISPlay	Turns on or off all bit displays of the specified pod of the logic input.	5-129
:LABel	Sets or queries the label of a bit of the specified pod of the logic input.	5-130
:LEVel	Sets or queries the user-defined threshold level of a bit of the specified pod of the logic input.	5-130
:TYPE	Selects the threshold level of a bit of the specified pod of the logic input.	5-130
:BUS?	Queries all logic input bus settings.	5-130
:BUS		
:ASSignment	Sets or queries the logic input bus assignment.	5-130
:DISPlay	Sets or queries the on/off state of the logic input bus display.	5-130
:FORMat	Sets or queries the logic input bus display format.	5-130
:LABel	Sets or queries the label of the logic input bus label.	5-130
:DESKew	Sets or queries the deskewing of the specified pod of the logic input.	5-130
:HYSTeresis	Sets or queries the hysteresis of the specified pod of the logic input.	5-131
:STATe?	Queries all settings related to the logic input state display.	5-131
:STATe		
:ASSignment?	Queries the logic input state display assignment.	5-131
:ASSignment		
:ALL	Sets or queries all settings related to the logic input state display assignment.	5-131
:BIT<x>	Sets or queries the logic input state display assignments of all bits.	5-131
:BUS	Sets or queries the bus setting of the logic input state display assignment.	5-131
:		
[:PODA]		
:STATe		
:CLOCK	Sets or queries the state display reference clock waveform of the logic input.	5-131
:HYSTeresis	Sets or queries the hysteresis of the state display reference clock waveform of the logic input.	5-131
:MODE	Sets or queries the on/off state of the logic input state display.	5-131
:POLarity	Sets or queries the polarity of the state display reference clock waveform of the logic input.	5-132
:THReshold	Sets or queries the detection level of the state display reference clock waveform of the logic input.	5-132
:POSition	Sets or queries the vertical position of the logic signal.	5-132
:SIZE	Sets or queries the display size of the logic signal.	5-132

5.1 List of Commands

Command	Function	Page
MATH Group		
:MATH<x>?	Queries all computation settings.	5-133
:MATH<x>		
:DISPlay	Sets or queries whether to show (ON) or hide (OFF) the computed waveforms.	5-133
:ECount?	Queries all edge count settings.	5-133
:ECount		
:HYSTeresis	Sets or queries the edge detection level hysteresis for edge count.	5-133
:POLarity	Sets or queries the edge detection polarity for edge count.	5-133
:THReshold	Sets or queries the edge detection Level for edge count computation.	5-133
:FILTer?	Queries all filter settings.	5-133
:FILTer		
:FORDer	Sets or queries the IIR filter's filter order.	5-133
:HCUToff	Sets or queries the cutoff frequency of the IIR filter high-pass filter.	5-133
:LCUToff	Sets or queries the cutoff frequency of the IIR filter low-pass filter.	5-134
:TIME	Sets or queries the phase shift delay.	5-134
:TYPE	Sets or queries the filter type.	5-134
:WEIGHt	Sets or queries the moving average weight.	5-134
:INTegral?	Queries all integral settings.	5-134
:INTegral		
:SPoint	Sets or queries the integral start position.	5-134
:LABel?	Queries all settings related to the label of the computed waveform.	5-134
:LABel		
[:DEFine]	Sets or queries a computed waveform label.	5-134
:MODE	Sets or queries the label display on/off state of computed waveforms.	5-134
:OPERation	Sets or queries the operator.	5-135
:RCount?	Queries all rotary count operation settings.	5-135
:RCount		
:COUNT?	Queries all count condition settings for the specified channel's rotary count operation.	5-135
:COUNT		
:CWTiming<y>	Sets or queries the clockwise (CW) count timing for rotary count operation.	5-135
:MULTi	Sets or queries the multiplication for rotary count operation.	5-135
:HYSTeresis<y>	Sets or queries the hysteresis for rotary count operation.	5-135
:RESet?	Queries all reset condition settings for the specified channel's rotary count operation.	5-135
:RESet		
:CWTiming	Sets or queries the clockwise (CW) reset timing for rotary count operation.	5-135
:RTIMing	Sets or queries the reset timing for rotary count operation.	5-135
:THReshold<y>	Sets or queries the determination level for rotary count operation.	5-136
:ZPOLarity	Sets or queries the polarity for rotary count operation.	5-136
:SCALE?	Queries all scaling settings.	5-136
:SCALE		
:CENTer	Sets or queries the center value for manual scaling.	5-136
:MODE	Sets or queries the scale mode.	5-136
:SENSitivity	Sets or queries the sensitivity for manual scaling.	5-136
:UNIT?	Queries all computation unit settings.	5-136
:UNIT		
[:DEFine]	Sets or queries the computation unit.	5-136
:MODE	Sets or queries whether a computation unit will be attached automatically or manually.	5-136
:USERdefine?	Queries all user-defined computation settings.	5-136
:USERdefine		
:AVERage?	Queries all settings related to the averaging of user-defined computation.	5-136
:AVERage		
:EWEight	Sets or queries the attenuation constant of exponential averaging of user-defined computation.	5-137
:MODE	Sets or queries the averaging mode for user-defined computation.	5-137
:CONStant<y>	Sets or queries a constant for user-defined computation.	5-137
:DEFine	Sets or queries an expression for user-defined computation.	5-137
:FILTer<y>?	Queries all settings related to the filter of user-defined computation.	5-137

5.1 List of Commands

Command	Function	Page
:FILTer<y>		
:BAND	Sets or queries the filter band (frequency band) for user-defined computation.	5-137
:CUTOff<z>	Sets or queries an cutoff frequency for user-defined computation.	5-137
:TYPE	Sets or queries the filter type for user-defined computation.	5-138
:HISTory		
:ABORt	Aborts the user-defined computation of history waveforms (Math on History).	5-138
:EXECute	Executes the user-defined computation of history waveforms (Math on History).	5-138
:SCALe?	Queries all settings related to the scaling of user-defined computation.	5-138
:SCALe		
:ARANGing	Executes auto range on user-defined computation.	5-138
:CENTer	Sets or queries the center value of scaling for user-defined computation.	5-138
:SENSitivity	Sets or queries the scaling sensitivity for user-defined computation.	5-138

MEASure Group

:MEASure?	Queries all the settings for automated measurement of waveform parameters.	5-139
:MEASure		
:BIT<x>?	Queries all the parameter on/off states of the specified logic waveform.	5-139
:BIT<x>		
:ALL	Collectively turns on or off all measurement items of a logic waveform.	5-139
:AREA2?	Queries all the parameter on/off states of a logic waveform in Area2.	5-139
:AREA2		
:ALL	Collectively turns on or off all measurement items of a logic waveform in Area2.	5-139
:<Parameter>?	Queries a waveform parameter setting of a logic waveform in Area2.	5-139
:<Parameter>		
:COUNT?	Queries the statistical processing count of a logic waveform parameter in Area2.	5-139
:{MAXimum MEAN MINimum SDEVIation}?	Queries a statistical value of a logic waveform parameter in Area2.	5-139
:STATe	Sets or queries the on/of state of a logic waveform parameter in Area2.	5-139
:VALue?	Queries automated measurement values of waveform parameters of a logic waveform in Area2.	5-140
:COPY	Copies the on/off state of all measurement items of a logic waveform to all other waveforms in Area2.	5-140
:DELay?	Queries all settings related to the delay between channels of a logic waveform in Area2.	5-140
:DELay		
:MEASure?	Queries all of the source waveform settings for measuring the delay between channels of a logic waveform in Area2.	5-140
:MEASure		
:COUNT	Sets or queries the count number of a source waveform for measuring the delay between channels of a logic waveform in Area2.	5-140
:SLOPe	Sets or queries the slope of a source waveform for measuring the delay between channels of a logic waveform in Area2.	5-140
:REFerence?	Queries all of the reference waveform settings for measuring the delay between channels of a logic waveform in Area2.	5-140
:REFerence		
:COUNT	Sets or queries the count number of the reference waveform for measuring the delay between channels of a logic waveform in Area2.	5-141
:SLOPe	Sets or queries the slope of the reference waveform for measuring the delay between channels of a logic waveform in Area2.	5-141
:SOURce	Sets or queries whether the reference point for measuring the delay between channels of a logic waveform will be set to the trigger point or a waveform in Area2.	5-141
:TRACe	Sets or queries the edge of the reference waveform for measuring the delay between channels of a logic waveform in Area2.	5-141
:STATe	Sets or queries the on/off state and display format of the delay parameter in Area2.	5-141
:<Parameter>?	Queries a waveform parameter setting of a logic waveform.	5-141
:<Parameter>		
:COUNT?	Queries the statistical processing count of a logic waveform parameter.	5-141

5.1 List of Commands

Command	Function	Page
: {MAXimum MEAN MINimum SDEViation}?	Queries a statistical value of a logic waveform parameter.	5-142
:STATe	Sets or queries the on/of state of a logic waveform parameter.	5-142
:VALue?	Queries automated measurement values of waveform parameters of a logic waveform.	5-142
:COPY	Copies the on/off state of all measurement items of a logic waveform to all other waveforms.	5-142
:DELAy?	Queries all settings related to the delay between channels of a logic waveform.	5-142
:DELAy		
:MEASure?	Queries all the settings for a source waveform for measuring the delay between channels of a logic waveform.	5-142
:MEASure		
:COUNT	Sets or queries the source waveform count for measuring the delay between channels of a logic waveform.	5-143
:SLOPe	Sets or queries a source waveform slope that will be used to measure delay between channels for a logic waveform.	5-143
:REFerence?	Queries all the settings for the reference waveform for measuring the delay between channels of a logic waveform.	5-143
:REFerence		
:COUNT	Sets or queries the reference waveform count for measuring the delay between channels of a logic waveform.	5-143
:SLOPe	Sets or queries the reference waveform slope that will be used to measure delay between channels for a logic waveform.	5-143
:SOURce	Sets or queries whether the reference point for measuring the delay between channels of a logic waveform will be set to the trigger point or a logic waveform.	5-143
:TRACe	Sets or queries the reference waveform edge that will be used to measure delay between channels for a logic waveform.	5-143
:STATe	Sets or queries the on/off state and display format of the delay parameter.	5-144
: {CHANnel<x> MATH<x>}?	Queries all the parameter on/off states of a waveform.	5-144
: {CHANnel<x> MATH<x>}		
:ALL	Collectively turns on or off all measurement items of a waveform.	5-144
:AREA2?	Queries all the parameter on/off states of a waveform in Area2.	5-144
:AREA2		
:ALL	Collectively turns on or off all measurement items of a waveform in Area2.	5-144
:<Parameter>?	Queries a waveform parameter setting of a waveform in Area2.	5-144
:<Parameter>		
:COUNT?	Queries the statistical processing count of a waveform parameter in Area2.	5-144
: {MAXimum MEAN MINimum SDEViation}?	Queries a statistical value of a waveform parameter in Area2.	5-145
:STATe	Sets or queries the on/of state of a waveform parameter in Area2.	5-145
:VALue?	Queries the automated measurement value of a waveform parameter in Area2.	5-145
:COPY	Copies the on/off state of all measurement items of a waveform to all other waveforms in Area2.	5-145
:CYCLe	Sets or queries cycle mode of Area2.	5-145
:DELAy?	Queries all settings related to the delay between channels of a waveform in Area2.	5-146
:DELAy		
:MEASure?	Queries all of the source waveform settings for measuring the delay between channels of a waveform in Area2.	5-146
:MEASure		
:COUNT	Sets or queries the count number of a source waveform for measuring the delay between channels of a waveform in Area2.	5-146
:SLOPe	Sets or queries the slope of a source waveform for measuring the delay between channels of a waveform in Area2.	5-146
:REFerence?	Queries all of the reference waveform settings for measuring the delay between channels of a waveform in Area2.	5-146
:REFerence		
:COUNT	Sets or queries the count number of the reference waveform for measuring the delay between channels of a waveform in Area2.	5-146
:SLOPe	Sets or queries the slope of the reference waveform for measuring the delay between channels of a waveform in Area2.	5-146

5.1 List of Commands

Command	Function	Page
:SOURCE	Sets or queries whether the reference point for measuring the delay between channels of a waveform will be set to the trigger point or a waveform in Area2.	5-147
:TRACE	Sets or queries the edge of the reference waveform for measuring the delay between channels of a waveform in Area2.	5-147
:STATE	Sets or queries the on/off state and display format of the delay parameter in Area2.	5-147
:<Parameter>?	Queries a waveform parameter setting of a waveform.	5-147
:<Parameter>		
:COUNT?	Queries the statistical processing count of a waveform parameter.	5-147
:{MAXimum MEAN MINimum S DEVIation}?	Queries a statistical value of a waveform parameter.	5-147
:STATE	Sets or queries the on/of state of a waveform parameter.	5-148
:VALUE?	Queries automated measurement values of waveform parameters of a waveform.	5-148
:COPY	Copies the on/off state of all measurement items of a waveform to all other waveforms.	5-148
:CYCLE	Sets or queries the cycle mode.	5-149
:DELAY?	Queries all settings related to the delay between channels of a waveform.	5-149
:DELAY		
:MEASURE?	Queries all the settings for a source waveform for measuring the delay between channels of a waveform.	5-149
:MEASURE		
:COUNT	Sets or queries the number of edges at which delay between channels will be measured for a waveform.	5-149
:SLOPE	Sets or queries a source waveform slope that will be used to measure delay between channels for a waveform.	5-149
:REFERENCE?	Queries all reference waveform settings used to measure the delay between channels for the specified waveform.	5-149
:REFERENCE		
:COUNT	Sets or queries the reference waveform count for measuring the delay between channels of a waveform.	5-149
:SLOPE	Sets or queries the reference waveform slope that will be used to measure delay between channels for a waveform.	5-150
:SOURCE	Sets or queries whether to set the reference point for measuring the delay between channels for a waveform to a trigger point or to a waveform.	5-150
:TRACE	Sets or queries the reference waveform edge that will be used to measure delay between channels for a waveform.	5-150
:STATE	Sets or queries the on/off state and display format of the delay parameter.	5-150
:DPROximal?	Queries all distal, mesial, and proximal settings.	5-150
:DPROximal		
:MODE	Sets or queries the distal, mesial, and proximal point mode setting.	5-150
:PERCent	Sets or queries the distal, mesial, and proximal points as percentages.	5-151
:UNIT	Sets or queries the distal, mesial, and proximal points as voltages.	5-151
:METHOD	Sets or queries the high and low points.	5-151
:CONTinuous?	Queries all settings for the normal statistical processing of automatically measured waveform parameters.	5-151
:CONTinuous		
:REStArt	Restarts the normal statistical processing of automatically measured waveform parameters.	5-151
:TLCHange	Sets or queries whether normal statistical processing of automatically measured waveform parameters is restarted when the trigger level is changed.	5-151
:CYCLE?	Queries all settings related to the cyclic statistical processing of automated measurement.	5-151
:CYCLE		
:ABORt	Aborts the execution of the cyclic statistical processing of automated measurement.	5-151
:EXECute	Executes the cyclic statistical processing of automated measurement.	5-151
:TRACe	Sets or queries the source waveform of the cyclic statistical processing of automated measurement.	5-151
:HISTory		
:ABORt	Aborts the execution of the statistical processing of the history waveform of automated measurement.	5-152

5.1 List of Commands

Command	Function	Page
:EXECute	Executes the statistical processing of the history waveform of automated measurement.	5-152
:INDicator	Sets or queries the measurement location indicator.	5-152
:MODE	Sets or queries the automated measurement mode.	5-152
:RANGe<x>	Sets or queries the measurement source window.	5-152
:TRANge<x>	Sets or queries the measurement time period.	5-152
:USER<x>?	Queries all automatic measurement settings for a Calc item.	5-152
:USER<x>		
:COUNT?	Queries the statistical processing count of the automatically measured value of a Calc item.	5-152
:DEFine	Sets or queries the expression for the automatically measured value of a Calc item.	5-152
: {MAXimum MEAN MINimum SDEViation}?	Queries a statistical value that is calculated on the automatically measured value of a Calc item.	5-152
:USER<X>		
:NAME	Sets or queries the name of a Calc item.	5-153
:USER<x>		
:STATe	Sets or queries the on/off status of automated measurement of a Calc item.	5-153
:UNIT	Sets or queries the unit of a Calc item.	5-153
:VALue?	Queries the automatically measured value of a Calc item.	5-153
:WAIT?	Waits for the completion of automated measurement with a timeout.	5-153
:ZLINKage	Sets or queries the linkage (on/off) between the measurement numbers and zoom positions of the results of automated measurement of waveform parameters.	5-153

RECall group

:RECall		
:SETup<x>		
:EXECute	Recalls setup data from the internal memory.	5-154

REFERENCE Group

:REFerence<x>?	Queries all reference waveform settings.	5-155
:REFerence<x>		
:DISPlay	Sets or queries the reference waveform display on/off state.	5-155
:LABel?	Queries all settings related to the label of a reference.	5-155
:LABel		
[:DEFine]	Sets or queries the label of a reference.	5-155
:MODE	Sets or queries the label display on/off state of a reference.	5-155
:LOAD	Loads a reference waveform.	5-155
:POSition	Sets or queries the vertical position of the reference waveform.	5-155

SEARCh Group

:SEARCh?	Queries all waveform search settings.	5-156
:SEARCh		
:ABORt	Aborts the search.	5-156
:ASCRoll<x>?	Queries all auto scroll settings.	5-156
:ASCRoll<x>		
:JUMP	Moves the center position of the zoom box to the left or right edge of the main window.	5-156
:SPEed	Sets or queries the auto scroll speed of the zoom box.	5-156
:STARt	Starts auto scrolling.	5-156
:STOP	Stops auto scrolling.	5-156
:EDGE?	Queries all edge search settings.	5-156
:EDGE		
:HYSteresis	Sets or queries the edge search level hysteresis.	5-156
:LEVel	Sets or queries the edge search level.	5-156
:SLOPe	Sets or queries the edge search slope.	5-156
:SOURce	Sets or queries the source waveform of edge search.	5-157
:EPOint	Sets or queries the search end position.	5-157
:EXECute	Executes a search. This is an overlap command.	5-157
:MARK	Sets or queries the detected point mark on/off state.	5-157

5.1 List of Commands

Command	Function	Page
:MAG<x>	Sets or queries the zoom factor of the zoom window.	5-157
:POSition<x>	Sets or queries the position of a zoom box.	5-157
:PATtern?	Queries all settings related to the combination (pattern) search of multiple inputs.	5-157
:PATtern		
:BITS?	Queries all settings related to the bits of the combination (pattern) search of multiple inputs.	5-157
:BITS		
:HEXa	Sets the bit pattern for the combination (pattern) search of multiple inputs in hexadecimal notation.	5-157
:PATtern	Sets or queries the bit pattern for the combination (pattern) search of multiple inputs in binary notation.	5-157
:{CHANnel<x> MATH<x>}?	Queries all settings related to a waveform of the combination (pattern) search of multiple inputs.	5-157
:{CHANnel<x> MATH<x>}		
:HYSTeresis	Sets or queries the hysteresis of the source waveform for the combination (pattern) search of multiple inputs.	5-158
:LEVel	Sets or queries the search level of the source waveform for the combination (pattern) search of multiple inputs.	5-158
:PATtern	Sets or queries the search pattern of the source waveform for the combination (pattern) search of multiple inputs.	5-158
:CLOCK?	Queries all settings related to the clock channel of the combination (pattern) search of multiple inputs.	5-158
:CLOCK		
:HYSTeresis	Sets or queries the hysteresis of the clock channel for the combination (pattern) search of multiple inputs.	5-158
:LEVel	Sets or queries the search level of the clock channel for the combination (pattern) search of multiple inputs.	5-158
:SLOPe	Sets or queries the slope of the clock channel for the combination (pattern) search of multiple inputs.	5-158
:SOURce	Sets or queries the source waveform of the clock channel for the combination (pattern) search of multiple inputs.	5-159
:CONDition	Sets or queries the achievement condition of the source waveform for the combination (pattern) search of multiple inputs.	5-159
:LOGic	Sets or queries the logic of the source waveform for the combination (pattern) search of multiple inputs.	5-159
:TIME<x>	Sets or queries the reference time of the source waveform for the combination (pattern) search of multiple inputs.	5-159
:TQQualify	Sets or queries the time condition of the source waveform for the combination (pattern) of multiple inputs.	5-159
:PULSe?	Queries all pulse width search settings.	5-159
:PULSe		
:HYSTeresis	Sets or queries the pulse width search level hysteresis.	5-159
:LEVel	Sets or queries the pulse width search level.	5-159
:POLarity	Sets or queries the pulse width search level polarity.	5-159
:SOURce	Sets or queries the source waveform of pulse width search.	5-159
:TIME<x>	Sets or queries the reference time of pulse width search.	5-160
:TQQualify	Sets or queries the time condition for pulse width searching.	5-160
:SElect	Queries the setting of the detected point to show in the zoom window and the zoom position of that point.	5-160
:SElect? MAXimum	Queries the total number of detected points.	5-160
:SKIP?	Queries all skip mode settings.	5-160
:SKIP		
:DECimation	Sets or queries the decimation detection of skip mode.	5-160
:HOLDoff	Sets or queries the hold-off detection.	5-160
:MODE	Sets or queries the mode of skip mode.	5-160
:SPOint	Sets or queries the search start position.	5-160
:TIMEout?	Queries all timeout period search settings.	5-160
:TIMEout		
:POLarity	Sets or queries the timeout period search polarity.	5-160
:SOURce	Sets or queries the timeout period search source waveform.	5-161
:TIME	Sets or queries the timeout period of the timeout period search.	5-161
:TWINDow	Sets or queries the zoom window for showing the detected area.	5-161

5.1 List of Commands

Command	Function	Page
:TYPE	Sets or queries the search type.	5-161
SERialbus group		
:SERialbus<x>?	Queries all serial bus signal analysis and search settings.	5-162
:SERialbus<x>		
:ASETup?	Queries all auto setup settings of serial bus signal analysis.	5-162
:ASETup		
:ABORt	Aborts the serial bus signal auto setup.	5-162
:EXECute	Execute an serial bus signal auto setup.	5-162
:SWAVeform	Sets or queries the waveform on which to perform serial bus signal auto setup.	5-162
:TSETing	Sets or queries whether trigger settings are included in the serial bus signal auto setup.	5-162
:CAN?	Queries all CAN bus signal settings.	5-162
:CAN		
:ANALyze?	Queries all CAN bus signal analysis settings.	5-163
[:ANALyze]		
:SETup?	Queries all CAN bus signal analysis bus settings.	5-163
:SETup		
:BRATe	Sets or queries the CAN bus signal analysis bit rate (data transfer rate).	5-163
:RECCessive	Sets or queries the CAN bus signal analysis recessive level.	5-163
:SOURce	Sets or queries the CAN bus signal analysis source.	5-163
:SPOint	Sets or queries the CAN bus signal analysis sample point.	5-163
:DETail?	Queries all CAN bus signal analysis result list settings.	5-163
:DETail		
:DISPlay	Sets or queries the display mode for the CAN bus signal analysis result list.	5-163
:LIST		
:ITEM?	Queries all items that will be displayed in the CAN bus signal analysis result list.	5-164
:VALue?	Queries all the data for the specified analysis number in the CAN bus signal analysis result list.	5-164
:SEARch?	Queries all CAN bus signal search settings.	5-164
:SEARch		
:ABORt	Aborts the CAN bus signal search.	5-164
:EXECute	Executes a CAN bus signal search.	5-164
:FJUMp		
:ACK	Jumps to the ACK field in the CAN bus signal search result.	5-164
:CONTRol	Jumps to the control field in the CAN bus signal search result.	5-164
:CRC	Jumps to the CRC field in the CAN bus signal search result.	5-164
:DATA	Jumps to the data field in the CAN bus signal search result.	5-164
:IDENtifier	Jumps to the identifier in the CAN bus signal search result.	5-164
:SOF	Jumps to the SOF in the CAN bus signal search result.	5-164
:SELEct	Sets which detected point to display in the CAN bus signal search zoom window and queries the zoom position of the detected point.	5-165
:SELEct? MAXimum	Queries the number of detected points in the CAN bus signal search.	5-165
:SETup?	Queries all CAN bus signal search condition settings.	5-165
:SETup		
:EFrame?	Queries all CAN bus signal search error settings.	5-165
:EFrame		
[:MODE]	Sets or queries the CAN bus signal search error frame setting.	5-165
:CRC	Sets or queries the CAN bus signal search CRC error setting.	5-165
:STUFF	Sets or queries the CAN bus signal search stuff error setting.	5-165
:IDData?	Queries all ID and data condition settings for CAN bus signal searching.	5-165
[:IDData]		
:ACK?	Queries all ID and data condition ACK settings for CAN bus signal searching.	5-166
:ACK		
:MODE	Sets or queries the ACK mode, which is one of the ID and data conditions, for CAN bus signal searching.	5-166
:TYPE	Sets or queries the ACK condition, which is one of the ID and data conditions, for CAN bus signal searching.	5-166
:DATA?	Queries all data settings for the ID and data conditions for CAN bus signal searching.	5-166

5.1 List of Commands

Command	Function	Page
:DATA		
:CONDition	Sets or queries the data comparison condition, which is one of the ID and data conditions, for CAN bus signal searching.	5-166
:DECimal<y>	Sets or queries a reference value, which is one of the ID and data conditions, for CAN bus signal searching in decimal notation.	5-166
:DLC	Sets or queries the number of significant bytes (DLC), which is one of the ID and data conditions, for CAN bus signal searching.	5-166
:DPATtern?	Queries all judgment data settings for the ID and data conditions for CAN bus signal searching.	5-167
:DPATtern		
:HEXa	Sets a reference value, which is one of the ID and data conditions, for CAN bus signal searching in hexadecimal notation.	5-167
:PATtern	Sets or queries the data, which is one of the ID and data conditions, for CAN bus signal searching in binary notation.	5-167
:ENDian	Sets or queries the byte order of the reference values, which is one of the ID and data conditions, for CAN bus signal searching.	5-167
:MODE	Sets or queries the data condition (enabled/disabled), which is one of the ID and data conditions, for CAN bus signal searching.	5-167
:MSBLsb	Sets or queries the MSB and LSB bits for the data, which is one of the ID and data conditions, for CAN bus signal searching.	5-167
:SIGN	Sets or queries the sign, which is one of the ID and data conditions, for CAN bus signal searching.	5-168
:IDENtifier?	Queries the ID, which is one of the ID and data conditions, for CAN bus signal searching.	5-168
:IDENtifier		
:HEXa	Sets an ID, which is one of the ID and data conditions, for CAN bus signal searching in hexadecimal notation.	5-168
:MFORmat	Sets or queries the ID frame format (standard or extended), which is one of the ID and data conditions, for CAN bus signal searching.	5-168
:MODE	Sets or queries the ID condition (enabled/disabled), which is one of the ID and data conditions, for CAN bus signal searching.	5-168
:PATtern	Sets or queries the ID condition, which is one of the ID and data conditions, for CAN bus signal searching in binary notation.	5-168
:PFORmat	Sets or queries the ID input format, which is one of the ID and data conditions, for CAN bus signal searching.	5-168
:MSIGnal?	Queries the message signal, which is one of the ID and data conditions, for CAN bus signal searching.	5-169
:MSIGnal		
:MESSage		
:ITEM	Sets the message item, which is one of the ID and data conditions, for CAN bus signal searching.	5-169
:SElect	Sets or queries the message signal condition, which is one of the ID and data conditions, for CAN bus signal searching.	5-169
:SIGnal?	Queries the signal, which is one of the ID and data conditions, for CAN bus signal searching.	5-169
:SIGnal		
:CONDition	Sets or queries the signal data condition, which is one of the ID and data conditions, for CAN bus signal searching.	5-169
:DECimal<y>	Sets or queries a reference signal value, which is one of the ID and data conditions, for CAN bus signal searching in decimal notation.	5-169
:ITEM	Sets the signal item, which is one of the ID and data conditions, for CAN bus signal searching.	5-169
:RTR	Sets or queries the RTR value, which is one of the ID and data conditions, for CAN bus signal searching.	5-170
:MODE	Sets or queries the CAN bus signal search mode setting.	5-170
:CANFD?	Queries all CAN FD bus signal settings.	5-170
:CANFD		
:ANALyze?	Queries all CAN FD bus signal analysis settings.	5-170
[:ANALyze]		
:SETup?	Queries all CAN FD bus signal analysis bus settings.	5-170
:SETup		
:BRATe	Sets or queries the CAN FD bus signal analysis bit rate (data transfer rate).	5-170
:DBRate	Sets or queries the CAN FD bus signal analysis data bit rate (data transfer rate of data phase).	5-170

5.1 List of Commands

Command	Function	Page
:DSPoint	Sets or queries the CAN FD bus signal analysis data phase sample point.	5-171
:FDStandard	Sets or queries whether the CAN FD bus signal to be analyzed is an ISO standard signal.	5-171
:REcessive	Sets or queries the CAN FD bus signal analysis recessive level.	5-171
:SOURce	Sets or queries the CAN FD bus signal analysis source.	5-171
:SPoint	Sets or queries the CAN FD bus signal analysis sample point.	5-171
:DETail?	Queries all CAN FD bus signal analysis result list settings.	5-171
:DETail		
:DISPlay	Sets or queries the display mode for the CAN FD bus signal analysis result list.	5-171
:LIST		
:ITEM?	Queries all items that will be displayed in the CAN FD bus signal analysis result list.	5-172
:VALue?	Queries all the data for the specified analysis number in the CAN FD bus signal analysis result list.	5-172
:SEARch?	Queries all CAN FD bus signal search settings.	5-172
:SEARch		
:ABORt	Aborts the CAN FD bus signal search.	5-172
:EXECute	Executes a CAN FD bus signal search.	5-172
:FJUMp		
:ACK	Jumps to the ACKField in the CAN FD bus signal search result.	5-172
:CONTRol	Jumps to the control field in the CAN FD bus signal search result.	5-172
:CRC	Jumps to the CRC field in the CAN FD bus signal search result.	5-172
:DATA	Jumps to the data field in the CAN FD bus signal search result.	5-172
:IDENtifier	Jumps to the identifier in the CAN FD bus signal search result.	5-172
:SOF	Jumps to the SOF in the CAN FD bus signal search result.	5-172
:SELect	Sets which detected point to display in the CAN FD bus signal search zoom window and queries the zoom position of the detected point.	5-173
:SELect? MAXimum	Queries the number of detected points in the CAN FD bus signal search.	5-173
:SETup?	Queries all CAN FD bus signal search condition settings.	5-173
:SETup		
:EFrame?	Queries all CAN FD bus signal search error settings.	5-173
:EFrame		
:CRC	Sets or queries the CAN FD bus signal search CRC error setting.	5-173
:CRCEFactor?	Queries all CRC error factor settings for CAN FD bus signal searching.	5-173
:CRCEFactor		
:CRCSequence	Sets or queries the CRC sequence (a CRC error factor) for CAN FD bus signal searching.	5-173
:SCount	Sets or queries the Stuff Count (a CRC error factor) for CAN FD bus signal searching.	5-174
:FSTuff	Sets or queries the CAN FD bus signal search fixed stuff error setting.	5-174
[:MODE]	Sets or queries the CAN FD bus signal search error frame setting.	5-174
:STUFF	Sets or queries the CAN FD bus signal search stuff error setting.	5-174
:FDF		
:CONDition	Sets or queries the CAN FD bus signal search FDF condition setting.	5-174
:IDData?	Queries all ID and data condition settings for CAN FD bus signal searching.	5-174
[:IDData]		
:ACK?	Queries all ACK settings for the ID and data conditions for CAN FD bus signal searching.	5-174
:ACK		
:MODE	Sets or queries the ACK mode, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-175
:TYPE	Sets or queries the ACK condition, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-175
:DATA?	Queries all data settings for the ID and data conditions for CAN FD bus signal searching.	5-175
:DATA		
:BCount	Sets or queries the pattern comparison position, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-175
:CONDition	Sets or queries the data comparison condition, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-175
:DBYTe	Sets or queries the number of data bytes, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-175

5.1 List of Commands

Command	Function	Page
:DECimal<y>	Sets or queries a reference value, which is one of the ID and data conditions, for CAN FD bus signal searching in decimal notation.	5-176
:DPATtern?	Queries all judgment data settings for the ID and data conditions for CAN FD bus signal searching.	5-176
:DPATtern		
:HEXa	Sets a reference value, which is one of the ID and data conditions, for CAN FD bus signal searching in hexadecimal notation.	5-176
:PATtern	Sets or queries the data value, which is one of the ID and data conditions, for CAN FD bus signal searching in binary notation.	5-176
:ENDian	Sets or queries the byte order of the reference values, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-176
:MODE	Sets or queries the data enable/disable condition, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-176
:MSBLsb	Sets or queries the MSB and LSB bits for the data, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-177
:SIGN	Sets or queries whether signed or unsigned data format will be used, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-177
:IDENtifier?	Queries all identifier settings for the ID and data conditions for CAN FD bus signal searching.	5-177
:IDENtifier		
:HEXa	Sets the ID value, which is one of the ID and data conditions, for CAN FD bus signal searching in hexadecimal notation.	5-177
:MFORmat	Sets or queries the ID frame format (standard or extended), which is one of the ID and data conditions, for CAN FD bus signal searching.	5-177
:MODE	Sets or queries the ID enable/disable condition, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-177
:PATtern	Sets or queries the ID pattern, which is one of the ID and data conditions, for CAN FD bus signal searching in binary notation.	5-177
:PFORmat	Sets or queries the ID input format, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-178
:MSIGnal?	Queries all message and signal settings for the ID and data conditions for CAN FD bus signal searching.	5-178
:MSIGnal		
:MESSage		
:ITEM	Sets the message item, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-178
:SElect	Sets or queries the message and signal condition, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-178
:SIGnal?	Queries all signal settings for the ID and data conditions for CAN FD bus signal searching.	5-178
:SIGnal		
:CONDition	Sets or queries the signal data condition, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-178
:DECimal<y>	Sets or queries a signal's reference value, which is one of the ID and data conditions, for CAN FD bus signal searching in decimal notation.	5-178
:ITEM	Sets the signal item, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-179
:RTR	Sets or queries the RTR value, which is one of the ID and data conditions, for CAN FD bus signal searching.	5-179
:MODE	Sets or queries the CAN FD bus signal search mode setting.	5-179
:CXPI?	Queries all CXPI bus signal analysis and search settings.	5-179
:CXPI		
:ANALyze?	Queries all CXPI bus signal analysis settings.	5-179
[:ANALyze]		
:SETup?	Queries all CXPI bus signal analysis bus settings.	5-179
:SETup		
:BRATe	Sets or queries the CXPI bus signal analysis bit rate (data transfer rate).	5-179
:CEDetection	Sets or queries the enable/disable condition of counter error detection for CXPI bus signal analysis.	5-179
:CTOLerance	Sets or queries the CXPI bus signal analysis clock tolerance.	5-180
:SOURce	Sets or queries the CXPI bus signal analysis source.	5-180
:TSAMple	Sets or queries the logic value (1 or 0) determination threshold for CXPI bus signal analysis.	5-180
:DETail?	Queries all CXPI bus signal analysis result list settings.	5-180

5.1 List of Commands

Command	Function	Page
:DETail		
:DISPlay	Sets or queries the display mode for the CXPI bus signal analysis result list.	5-180
:LIST		
:ALL?	Queries all the data for all analysis numbers in the CXPI bus signal analysis result list.	5-180
:ITEM?	Queries all items that will be displayed in the CXPI bus signal analysis result list.	5-180
:VALue?	Queries all the data for the specified analysis number in the CXPI bus signal analysis result list.	5-180
:SEARCh?	Queries all CXPI bus signal search settings.	5-181
:SEARCh		
:ABORt	Aborts the CXPI bus signal search.	5-181
:EXECute	Executes a CXPI bus signal search.	5-181
:SELEct	Sets which detected point to display in the CXPI bus signal search zoom window and queries the zoom position of the detected point.	5-181
:SETUp?	Queries all CXPI bus signal search condition settings.	5-181
:SETUp		
:ERRor?	Queries all CXPI bus signal search error settings.	5-181
:ERRor		
:CLOCk	Sets or queries the CXPI bus signal search clock error setting.	5-181
:COUNter	Sets or queries the CXPI bus signal search counter error setting.	5-181
:CRC	Sets or queries the CXPI bus signal search CRC error setting.	5-181
:DLENgth	Sets or queries the CXPI bus signal search data length error setting.	5-182
:FRAMing	Sets or queries the CXPI bus signal search framing error setting.	5-182
:IBS	Sets or queries the CXPI bus signal search IBS error setting.	5-182
:PARity	Sets or queries the CXPI bus signal search parity error setting.	5-182
:IDData?	Queries all ID and data condition settings for CXPI bus signal searching.	5-182
[:IDData]		
:DATA?	Queries all data settings for the ID and data conditions for CXPI bus signal searching.	5-182
:DATA		
:BCount	Sets or queries the pattern comparison position, which is one of the ID and data conditions, for CXPI bus signal searching.	5-182
:CONDition	Sets or queries the data comparison condition, which is one of the ID and data conditions, for CXPI bus signal searching.	5-183
:DBYTe	Sets or queries the number of data bytes, which is one of the ID and data conditions, for CXPI bus signal searching.	5-183
:DECimal<y>	Sets or queries a reference value, which is one of the ID and data conditions, for CXPI bus signal searching in decimal notation.	5-183
:DPATtern?	Queries all judgment data settings for the ID and data conditions for CXPI bus signal searching.	5-183
:DPATtern		
:HEXa	Sets a reference value, which is one of the ID and data conditions, for CXPI bus signal searching in hexadecimal notation.	5-183
:PATtern	Sets or queries a reference value, which is one of the ID and data conditions, for CXPI bus signal searching in binary notation.	5-183
:ENDian	Sets or queries the byte order of the reference values, which is one of the ID and data conditions, for CXPI bus signal searching.	5-184
:MODE	Sets or queries the ID and data condition (enabled/disabled) for CXPI bus signal searching.	5-184
:MSBLsb	Sets or queries the MSB and LSB bits for the data, which is one of the ID and data conditions, for CXPI bus signal searching.	5-184
:SIGN	Sets or queries the sign, which is one of the ID and data conditions, for CXPI bus signal searching.	5-184
:FINformation?	Queries all frame information settings for the ID and data conditions for CXPI bus signal searching.	5-184
:FINformation		
:CT	Sets or queries the frame information counter value, which is one of the ID and data conditions, for CXPI bus signal searching.	5-184
:MODE	Sets or queries the frame information (enabled/disabled), which is one of the ID and data conditions, for CXPI bus signal searching.	5-185
:SLEEP	Sets or queries the frame information sleep bit, which is one of the ID and data conditions, for CXPI bus signal searching.	5-185

5.1 List of Commands

Command	Function	Page
:WAKEup	Sets or queries the frame information wakeup bit, which is one of the ID and data conditions, for CXPI bus signal searching.	5-185
:ID?	Queries all ID settings for the ID and data conditions for CXPI bus signal searching.	5-185
:ID		
:HEXa	Sets an ID, which is one of the ID and data conditions, for CXPI bus signal searching in hexadecimal notation.	5-185
:MODE	Sets or queries the ID condition (enabled/disabled), which is one of the ID and data conditions, for CXPI bus signal searching.	5-185
:PATtern	Sets or queries the ID, which is one of the ID and data conditions, for CXPI bus signal searching in binary notation.	5-185
:PTYPE	Sets or queries the PTYPE comparison condition, which is one of the ID and data conditions, for CXPI bus signal searching.	5-186
:MODE	Sets or queries the CXPI bus signal search type.	5-186
:PTYPE	Queries all CXPI bus signal search PTYPE settings.	5-186
:CONDition	Sets or queries the CXPI bus signal search PTYPE condition setting.	5-186
:WAKEupsleep?	Queries all wakeup and sleep settings for CXPI bus signal searching.	5-186
:WAKEupsleep		
:SFRame	Sets or queries the CXPI bus signal search sleep frame setting.	5-186
:SLEEP	Sets or queries the CXPI bus signal search sleep (clock unavailable condition) setting.	5-186
:WAKEup	Sets or queries the CXPI bus signal search wakeup (clock available condition) setting.	5-187
:WPULse	Sets or queries the CXPI bus signal search wakeup pulse setting.	5-187
:DECode?	Queries all settings related to the decoding of serial bus signals.	5-187
:DECode		
[:FORMat]	Sets or queries the decode display format of serial bus signals.	5-187
:LABel	Sets or queries the decode label name of serial bus signals.	5-187
:POSition	Sets or queries the vertical position for decoding serial bus signals.	5-187
:SENT?	Queries all SENT signal decode settings.	5-187
:SENT		
:FAST?	Queries all settings for decoding the fast channel of SENT signals.	5-187
:FAST		
:LABel	Sets or queries the label name for decoding the fast channel of SENT signals.	5-187
:POSition	Sets or queries the vertical position for decoding the fast channel of SENT signals.	5-188
:SLOW?	Queries all settings for decoding the slow channel of SENT signals.	5-188
:SLOW		
:LABel	Sets or queries the label name for decoding the slow channel of SENT signals.	5-188
:POSition	Sets or queries the vertical position for decoding the slow channel of SENT signals.	5-188
:SPI?	Queries all SPI bus signal decode settings.	5-188
:SPI		
:DATA<y>?	Queries all settings related to the specified data for SPI bus signal decoding.	5-188
:DATA<y>		
:LABel	Sets or queries the label name for decoding SPI bus signals.	5-188
:POSition	Sets or queries the vertical position for decoding SPI bus signals.	5-189
:DISPlay	Sets or queries whether serial bus signal analysis is to be performed.	5-189
:FLEXray?	Queries all FlexRay bus signal analysis and search settings.	5-189
:FLEXray		
:ANALyze?	Queries all FlexRay bus signal analysis settings.	5-189
[:ANALyze]		
:SETup?	Queries all FlexRay bus signal analysis bus settings.	5-189
:SETup		
:BCHannel	Sets or queries the channel bus type for FlexRay bus signal analysis.	5-189
:BRATe	Sets or queries the FlexRay bus signal analysis bit rate (data transfer rate).	5-189
:SOURce	Sets or queries the FlexRay bus signal analysis source.	5-189
:DETail?	Queries all FlexRay bus signal analysis result list settings.	5-190
:DETail		
:DISPlay	Sets or queries the display mode for the FlexRay bus signal analysis result list.	5-190

5.1 List of Commands

Command	Function	Page
:LIST		
:ITEM?	Queries all items that will be displayed in the FlexRay bus signal analysis result list.	5-190
:VALue?	Queries all the data for the specified analysis number in the FlexRay bus signal analysis result list.	5-190
:SEARCh?	Queries all FlexRay bus signal search settings.	5-190
:SEARCh		
:ABORt	Aborts the FlexRay bus signal search.	5-190
:EXECute	Executes a FlexRay bus signal search.	5-190
:FJUMp		
:CCOutt	Jumps to the cycle count field in the FlexRay bus signal search result.	5-190
:CRC	Jumps to the CRC field in the FlexRay bus signal search result.	5-190
:FRAMeId	Jumps to the frame ID field in the FlexRay bus signal search result.	5-190
:HCRC	Jumps to the header CRC field in the FlexRay bus signal search result.	5-190
:PLENgtH	Jumps to the payload length field in the FlexRay bus signal search result.	5-191
:SELEct	Sets the detected waveform number for a FlexRay bus signal search and queries the zoom position corresponding to that number.	5-191
:SETUp?	Queries all FlexRay bus signal search condition settings.	5-191
:SETUp		
:ERRor?	Queries all FlexRay bus signal search error settings.	5-191
:ERRor		
:BSS	Sets or queries the FlexRay bus signal search BSS Error setting.	5-191
:CRC	Sets or queries the FlexRay bus signal search CRC Error setting.	5-191
:FES	Sets or queries the FlexRay bus signal search FES Error setting.	5-191
:HCRC	Sets or queries the FlexRay bus signal search Header CRC Error setting.	5-191
:IDData?	Queries all ID and data condition settings for FlexRay bus signal search.	5-192
[:IDData]		
:CCOutt?	Queries all FlexRay bus signal search cycle count settings.	5-192
:CCOutt		
:CONDition	Sets or queries the FlexRay bus signal search cycle count data condition.	5-192
:COUNT<y>	Sets or queries the FlexRay bus signal search cycle count setting.	5-192
:MODE	Sets or queries the cycle count mode, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-192
:DATA?	Queries all data settings for the ID and data conditions for FlexRay bus signal searching.	5-193
:DATA		
:BCOutt	Sets or queries the pattern comparison position, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-193
:CONDition	Sets or queries the data comparison condition, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-193
:DBYTe	Sets or queries the number of data bytes, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-193
:DECimal<y>	Sets or queries a reference value, which is one of the ID and data conditions, for FlexRay bus signal searching in decimal notation.	5-193
:DPATtern?	Queries all judgment data settings for the ID and data conditions for FlexRay bus signal searching.	5-193
:DPATtern		
:HEXa	Sets a reference value, which is one of the ID and data conditions, for FlexRay bus signal searching in hexadecimal notation.	5-194
:PATtern	Sets or queries the data, which is one of the ID and data conditions, for FlexRay bus signal searching in binary notation.	5-194
:ENDian	Sets or queries the byte order of the reference values, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-194
:MODE	Sets or queries the data condition (enabled/disabled), which is one of the ID and data conditions, for FlexRay bus signal searching.	5-194
:MSBLsb	Sets or queries the MSB and LSB bits for the data, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-194
:SIGN	Sets or queries the sign, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-194
:FID?	Queries the frame ID, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-195
:FID		

5.1 List of Commands

Command	Function	Page
:CONDition	Sets or queries the frame ID data condition, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-195
:ID<y>	Sets or queries the frame ID value, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-195
:MODE	Sets or queries the frame ID condition (enabled/disabled), which is one of the ID and data conditions, for FlexRay bus signal searching.	5-195
:INDicator?	Queries the indicator, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-195
:INDicator		
:MODE	Sets or queries the indicator condition (enabled/disabled), which is one of the ID and data conditions, for FlexRay bus signal searching.	5-195
:NFRame	Sets or queries the indicator null frame, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-196
:PPReamble	Sets or queries the indicator payload preamble, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-196
:STFRame	Sets or queries the indicator start frame, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-196
:SYFRame	Sets or queries the indicator sync frame, which is one of the ID and data conditions, for FlexRay bus signal searching.	5-196
:MODE	Sets or queries the FlexRay bus signal search mode setting.	5-196
:I2C?	Queries all I2C bus signal analysis and search settings.	5-196
:I2C		
:ANALyze?	Queries all I2C bus signal analysis settings.	5-196
[:ANALyze]		
:SETup?	Queries all I2C bus signal analysis bus settings.	5-196
:SETup		
:CLOCK?	Queries all I2C bus signal analysis clock settings.	5-197
:CLOCK		
:SOURce	Sets or queries the I2C bus signal analysis clock.	5-197
:DATA?	Queries all I2C bus signal analysis data settings.	5-197
:DATA		
:SOURce	Sets or queries the I2C bus signal analysis data.	5-197
:INCLuderw	Sets or queries the enable/disable condition of address R/W for I2C bus signal analysis.	5-197
:DETail?	Queries all I2C bus signal analysis result list settings.	5-197
:DETail		
:DISPlay	Sets or queries the display mode for the I2C bus signal analysis result list.	5-197
:LIST		
:ITEM?	Queries all items that will be displayed in the I2C bus signal analysis result list.	5-197
:VALue?	Queries all the data for the specified analysis number in the I2C bus signal analysis result list.	5-198
:SEARch?	Queries all I2C bus signal search settings.	5-198
:SEARch		
:ABORt	Aborts the I2C bus signal search.	5-198
:EXECute	Executes an I2C bus signal search.	5-198
:SElect	Sets which detected point to display in the I2C bus signal search zoom window and queries the zoom position of the detected point.	5-198
:SETup?	Queries all I2C bus signal search condition settings.	5-198
[:SETup]		
:ADDRess?	Queries all I2C bus signal search address pattern settings.	5-198
:ADDRess		
:ADDRess?	Queries all settings related to the address of the I2C bus signal search address pattern.	5-198
:ADDRess		
:BIT10Address?	Queries all I2C bus signal search 10-bit address settings.	5-198
:BIT10Address		
:DIRection	Sets or queries the 10-bit address data direction for I2C bus signal searching.	5-198
:HEXa	Sets the 10-bit address for I2C bus signal searching in hexadecimal notation.	5-199
:PATtern	Sets or queries the 10-bit address for I2C bus signal searching in binary notation.	5-199
ADDRess[:		

5.1 List of Commands

Command	Function	Page
:BIT7Address]? :BIT7Address]	Queries all I2C bus signal search 7-bit address settings.	5-199
:DIRection	Sets or queries the 7-bit address data direction for I2C bus signal searching.	5-199
:HEXa	Sets the 7-bit address for I2C bus signal searching in hexadecimal notation.	5-199
:PATtern	Sets or queries the 7-bit address for I2C bus signal searching in binary notation.	5-199
:ADDress		
:BIT7APsub?	Queries all I2C bus signal search 7-bit + sub address settings.	5-200
:BIT7APsub		
:ADDress?	Queries all settings related to the 7-bit address of the 7-bit + sub address for I2C bus signal searching.	5-200
:ADDress		
:DIRection	Sets or queries the 7-bit + sub address data direction for I2C bus signal searching.	5-200
:HEXa	Sets the 7-bit address of 7-bit + sub address for I2C bus signal searching in hexadecimal notation.	5-200
:PATtern	Sets or queries the 7-bit address of 7-bit + sub address for I2C bus signal searching in binary notation.	5-200
:SADDress?	Queries all settings related to the sub address of the 7-bit + sub address for I2C bus signal searching.	5-200
:SADDress		
:HEXa	Sets the sub address of 7-bit + sub address for I2C bus signal searching in hexadecimal notation.	5-200
:PATtern	Sets or queries the sub address of 7-bit + sub address for I2C bus signal searching in binary notation.	5-201
:MODE	Sets or queries the address condition enable/disable state for I2C bus signal searching.	5-201
:TYPE	Sets or queries the address condition address format for I2C bus signal searching.	5-201
:DATA?	Queries all settings related to the data of the I2C bus signal search address pattern.	5-201
:DATA		
:BCount	Sets or queries the data pattern comparison position for I2C bus signal searching.	5-201
:BMode	Sets or queries on/off state of the data pattern comparison position for I2C bus signal searching.	5-201
:CONDition	Sets or queries the data judgment method (true/false) for I2C bus signal searching.	5-202
:DBYte	Sets or queries the number of reference values for I2C bus signal searching.	5-202
:DPATtern?	Queries all data settings for the data conditions for I2C bus signal searching.	5-202
:DPATtern		
:HEXa	Sets the data condition data for I2C bus signal searching in hexadecimal notation.	5-202
:PATtern	Sets or queries the data condition data for I2C bus signal searching in binary notation.	5-202
:MODE	Sets or queries the data condition enable/disable state for I2C bus signal searching.	5-202
:GENeralcall?	Queries all I2C bus signal search general call settings.	5-202
:GENeralcall		
:BIT7Maddress?	Queries all settings related to the 7-bit master address of general call for I2C bus signal searching.	5-202
:BIT7Maddress		
:HEXa	Sets the 7-bit master address of general call for I2C bus signal searching in hexadecimal notation.	5-203
:PATtern	Sets or queries the 7-bit master address of general call for I2C bus signal searching in binary notation.	5-203
:SBYte	Sets or queries the general call second byte type for I2C bus signal searching.	5-203
:MODE	Sets or queries the I2C bus signal search mode setting.	5-203
:NONack?	Queries all I2C bus signal search NON ACK ignore mode settings.	5-203
:NONack		
:HSMode	Sets or queries whether NONACK is to be ignored in high-speed mode of I2C bus signal searching.	5-203

5.1 List of Commands

Command	Function	Page
:READaccess	Sets or queries whether NONACK is to be ignored in read access mode of I2C bus signal searching.	5-204
:STARTbyte	Sets or queries whether NONACK is to be ignored in the start byte of I2C bus signal searching.	5-204
:LIN?	Queries all LIN bus signal analysis and search settings.	5-204
:LIN		
:ANALyze?	Queries all LIN bus signal analysis settings.	5-204
[:ANALyze]		
:SETup?	Queries all LIN bus signal analysis bus settings.	5-204
:SETup		
:BRATe	Sets or queries the LIN bus signal analysis bit rate (data transfer rate).	5-204
:REVisiOn	Sets or queries the LIN bus signal analysis revision number.	5-204
:SOURce	Sets or queries the LIN bus signal analysis source.	5-204
:SPOint	Sets or queries the LIN bus signal analysis sample point.	5-205
:DETail?	Queries all LIN bus signal analysis result list settings.	5-205
:DETail		
:DISPlay	Sets or queries the display mode for the LIN bus signal analysis result list.	5-205
:LIST		
:ITEM?	Queries all items that will be displayed in the LIN bus signal analysis result list.	5-205
:LIST		
:VALue?	Queries all the data for the specified analysis number in the LIN bus signal analysis result list.	5-205
:SERialbus<x>		
:LIN		
:SEARch?	Queries all LIN bus signal search settings.	5-205
:SEARch		
:ABORt	Aborts the LIN bus signal search.	5-205
:EXECute	Executes a LIN bus signal search.	5-205
:FJUMp		
:BREAk	Jumps to the break field in the LIN bus signal search result.	5-205
:CSUM	Jumps to the checksum field in the LIN bus signal search result.	5-205
:DATA	Jumps to the data field in the LIN bus signal search result.	5-205
:IDENtifier	Jumps to the identifier field in the LIN bus signal search result.	5-206
:SYNCh	Jumps to the sync field in the LIN bus signal search result.	5-206
:SELEct	Sets which detected point to display in the LIN bus signal search zoom window and queries the zoom position of the detected point.	5-206
:SELEct? MAXimum	Queries the number of detected points in the LIN bus signal search.	5-206
:SETup?	Queries all LIN bus signal search condition settings.	5-206
:SETup		
:ERRor?	Queries all LIN bus signal search error settings.	5-206
:ERRor		
:CHECksum	Sets or queries the LIN bus signal search checksum error setting.	5-206
:FRAMing	Sets or queries the LIN bus signal search framing error setting.	5-206
:PARity	Sets or queries the LIN bus signal search parity error setting.	5-206
:SYNCh	Sets or queries the LIN bus signal search synch error setting.	5-207
:TIMeout	Sets or queries the LIN bus signal search timeout error setting.	5-207
:IDData?	Queries all ID and data condition settings for LIN bus signal searching.	5-207
:IDData		
:DATA?	Queries all LIN bus signal search data settings.	5-207
:DATA		
:CONDition	Sets or queries the LIN bus signal search data condition setting.	5-207
:DBYTe	Sets or queries the number of data bytes for LIN bus signal searching.	5-207
:DECimal<y>	Sets or queries the data for LIN bus signal searching in decimal notation.	5-207
:DPATtern?	Queries all LIN bus signal search data settings.	5-208
:DPATtern		
:HEXa	Sets the data for LIN bus signal searching in hexadecimal notation.	5-208
:PATtern	Sets or queries the data for LIN bus signal searching in binary notation.	5-208
:ENDian	Sets or queries the data endian for LIN bus signal searching.	5-208
:MODE	Sets or queries the data condition enable/disable state for LIN bus signal searching.	5-208
:MSBLsb	Sets or queries the MSB/LSB bit for LIN bus signal searching.	5-208

5.1 List of Commands

Command	Function	Page
:SIGN	Sets or queries the data sign for LIN bus signal searching.	5-208
:IDENtifier?	Queries all LIN bus signal search identifier settings.	5-209
:IDENtifier		
:ID?	Queries all LIN bus signal search ID settings.	5-209
:ID		
:HEXa	Sets the ID for LIN bus signal searching in hexadecimal notation.	5-209
:MODE	Sets or queries the ID condition enable/disable state for LIN bus signal searching.	5-209
:PATtern	Sets or queries the ID for LIN bus signal searching in binary notation.	5-209
:MODE	Sets or queries the LIN bus signal search mode setting.	5-209
:PSI5?	Queries all PSI5 signal analysis and search settings.	5-209
:PSI5		
:ANALyze?	Queries all PSI5 signal analysis settings.	5-209
[:ANALyze]		
:SETup?	Queries all bus setup settings for PSI5 signal analysis.	5-209
:SETup		
:DATA?	Queries all PSI5 signal analysis data signal settings.	5-210
:DATA		
:BRaTe	Sets or queries the PSI5 signal analysis bit rate (data transfer rate).	5-210
:CTOLerance	Sets or queries the PSI5 signal analysis clock tolerance range.	5-210
:DBITs	Sets or queries the data length for PSI5 signal analysis.	5-210
:EDEtECTION	Sets or queries the error detection method for PSI5 signal analysis.	5-210
:SNRejection?	Queries all PSI5 signal analysis sync noise rejection settings.	5-210
:SNRejection		
:ETImE	Sets or queries the rejection end time for PSI5 signal analysis sync noise rejection.	5-211
:MODE	Sets or queries the on/off state of PSI5 signal analysis sync noise rejection.	5-211
:SOURce	Sets or queries the data source for PSI5 signal analysis.	5-211
:NUMBerofslot	Sets or queries the number of slots for PSI5 signal analysis.	5-211
:SLOT<y>	Sets or queries the slot's start and end times for PSI5 signal analysis.	5-211
:SYNC	Sets or queries the sync signal for PSI5 signal analysis.	5-212
:DETail?	Queries all PSI5 signal analysis list settings.	5-212
:DETail		
:DISPlay	Sets or queries the display mode for the PSI5 signal analysis list.	5-212
:LIST		
:ALL?	Queries all the data for all analysis numbers in the PSI5 signal analysis result list.	5-212
:ITEM?	Queries all items that will be displayed in the PSI5 signal analysis result list.	5-212
:VALue?	Queries all the data for the specified analysis number in the PSI5 signal analysis result list.	5-212
:SEARCh?	Queries all PSI5 signal search settings.	5-212
:SEARCh		
:ABORt	Aborts the PSI5 signal search.	5-212
:EXECute	Executes a PSI5 signal search.	5-212
:SELect	Sets which detected point to display in the PSI5 signal search zoom window and queries the zoom position of the detected point.	5-213
:SETup?	Queries all PSI5 signal search setup settings.	5-213
[:SETup]		
:DATA?	Queries all PSI5 signal data search settings.	5-213
:DATA		
:DATA?	Queries all data settings of the PSI5 signal data search.	5-213
:DATA		
:CONDition	Sets or queries the comparison condition of the PSI5 signal data search.	5-213
:DECimal<y>	Sets or queries the data of the PSI5 signal data search in decimal notation.	5-213
:HEXa	Sets the data of the PSI5 signal data search in hexadecimal notation.	5-213
:PATtern	Sets or queries the data of the PSI5 signal data search in binary notation.	5-213
:FRAMeinslot?	Queries all slot specification settings of the PSI5 signal data search.	5-214
:FRAMeinslot		
:MODE	Sets or queries the slot specification condition (enabled/disabled) of the PSI5 signal data search.	5-214
:SNUMber	Sets or queries the slot number of the PSI5 signal data search.	5-214
:ERRor?	Queries all PSI5 signal error search settings.	5-214

5.1 List of Commands

Command	Function	Page
:ERRor		
:FNUMber	Sets or queries the PSI5 signal frame number error search.	5-214
:FRAMe	Sets or queries the PSI5 signal frame error search.	5-214
:MANChester	Sets or queries the PSI5 signal Manchester error search.	5-214
:PCRC	Sets or queries the PSI5 signal parity/CRC error search.	5-215
:SBIT	Sets or queries the PSI5 signal start bit error search.	5-215
:SBOundary	Sets or queries the PSI5 signal slot boundary error search.	5-215
:FRAMEinslot?	Queries all PSI5 signal slot specification search settings.	5-215
:FRAMEinslot		
:SNUMber	Sets or queries the slot number of the PSI5 signal slot specification search.	5-215
:MODE	Sets or queries the PSI5 signal search mode setting.	5-215
:TRENd<y>?	Queries all PSI5 signal analysis trend display settings.	5-215
:TRENd<y>		
:ASCale	Executes auto scaling of the PSI5 signal analysis trend display.	5-215
:CURSor?	Queries all PSI5 signal analysis trend display cursor measurement settings.	5-216
:CURSor		
:DT		
:VALue?	Queries the time value between cursors on the PSI5 signal analysis trend display.	5-216
:DV	Queries the vertical value between cursors on the PSI5 signal analysis trend display.	5-216
:MODE	Sets or queries the cursor mode on the PSI5 signal analysis trend display.	5-216
:POSition<z>	Sets or queries the position of the specified cursor on the PSI5 signal analysis trend display.	5-216
:T<z>		
:VALue?	Queries the time value at the specified cursor on the PSI5 signal analysis trend display.	5-216
:V<z>	Queries the vertical value at the specified cursor on the PSI5 signal analysis trend display.	5-216
:DISPlay	Sets or queries the on/off state of the PSI5 signal analysis trend display.	5-216
:HRANge	Sets or queries the PSI5 signal analysis trend display source window.	5-217
:SOURce	Sets or queries the target slot number of the PSI5 signal analysis trend display.	5-217
:VERTical	Sets or queries the vertical range of the PSI5 signal analysis trend display.	5-217
:VTDisplay	Sets or queries the on/off status of the VT waveform display on the PSI5 signal analysis trend display.	5-217
:RWINDow	Sets or queries the zoom window for showing the detected area.	5-217
:SENT?	Queries all SENT signal analysis and search settings.	5-217
:SENT		
:ANALyze?	Queries all SENT signal analysis settings.	5-217
[:ANALyze]		
:SETup?	Queries all bus setup settings for SENT signal analysis.	5-217
:SETup		
:DISPlay	Sets or queries the display mode for the SENT signal analysis result.	5-218
:FAST?	Queries all SENT signal analysis fast channel settings.	5-218
:FAST		
:DTYpe	Sets or queries the SENT signal analysis fast channel data format.	5-218
:USETup?	Queries all settings related to user-defined data of the SENT signal analysis fast channel.	5-218
:USETup		
:DATA<y>?	Queries all settings related to user-defined data of the SENT signal analysis fast channel.	5-218
:DATA<y>		
:MODE	Sets or queries the enable/disable condition of the specified user-defined data of the SENT signal analysis fast channel.	5-218
:ORDer	Sets or queries the byte order of the specified user-defined data of the SENT signal analysis fast channel.	5-218
:SIZE	Sets or queries the data size of the specified user-defined data of the SENT signal analysis fast channel.	5-219
:MULTiplexing	Sets or queries the multiplexing enable/disable condition of the specified user-defined data of the SENT signal analysis fast channel.	5-219
:FORMat?	Queries all SENT signal analysis format settings.	5-219
:FORMat		

5.1 List of Commands

Command	Function	Page
:CEFactor?	Queries all SENT signal analysis error factor settings.	5-219
:CEFactor		
:SAComm?	Queries all status and communication error factor settings of SENT signal analysis.	5-219
:SAComm		
:BIT<y>?	Sets or queries status and communication error factor bit of SENT signal analysis.	5-219
:SCPulses	Sets or queries the consecutive calibration pulse error factor of SENT signal analysis.	5-219
:CRCType	Sets or queries the SENT signal analysis CRC computation type.	5-219
:CTICK	Sets or queries the SENT signal analysis clock tick value.	5-220
:CTOLerance	Sets or queries the SENT signal analysis clock tolerance.	5-220
:DNIBbles	Sets or queries the number of data nibbles for SENT signal analysis.	5-220
:PPULse	Sets or queries the presence or absence of pause pulses for SENT signal analysis.	5-220
:VERSion	Sets or queries the SENT signal analysis specification version.	5-220
:STYPe	Sets or queries the SENT signal analysis slow channel format.	5-220
:SOURce	Sets or queries the SENT signal analysis source.	5-221
:DETail?	Queries all SENT signal analysis list settings.	5-221
:DETail		
:DISPlay	Sets or queries the display mode for the SENT signal analysis list.	5-221
:LIST		
:ALL?	Queries all the data for all analysis numbers in the SENT signal analysis result list.	5-221
:ITEM?	Queries all items that will be displayed in the SENT signal analysis result list.	5-221
:VALue?	Queries all the data for the specified analysis number in the SENT signal analysis result list.	5-221
:SEARCh?	Queries all SENT signal search settings.	5-221
:SEARCh		
:ABORt	Aborts the SENT signal search.	5-221
:EXECute	Executes a SENT signal search.	5-221
:SELEct	Sets which detected point to display in the SENT signal search zoom window and queries the zoom position of the detected point.	5-222
:SELEct? MAXimum	Queries the number of detected points in the SENT signal search.	5-222
:SETup?	Queries all SENT signal search setup settings.	5-222
[:SETup]		
:ERRor?	Queries all SENT signal search error settings.	5-222
:ERRor		
:FCRC	Sets or queries the SENT signal search fast channel CRC error.	5-222
:NDValue	Sets or queries the SENT signal search nibble data value error.	5-222
:NNUmber	Sets or queries the SENT signal search nibble data count error.	5-222
:SAComm	Sets or queries the SENT signal search status and communication error.	5-223
:SCPulses	Sets or queries the consecutive calibration pulse error for SENT signal searching.	5-223
:SCRC	Sets or queries the SENT signal search slow channel CRC error.	5-223
:FDATA?	Queries all fast channel data settings for SENT signal searching.	5-223
:FDATA		
:DATA<y>?	Queries all settings related to the specified fast channel user data for SENT signal searching.	5-223
:DATA<y>		
:CONDition	Sets or queries the comparison condition of the specified user data of the SENT signal search fast channel.	5-223
:DECimal<z>	Sets or queries the data of the specified user data of the SENT signal search fast channel in decimal notation.	5-223
:MODE	Sets or queries the data enable/disable condition of the specified user data of the SENT signal search fast channel.	5-224
:DNIBbles?	Queries all fast channel nibble data settings for SENT signal searching.	5-224
:DNIBbles		
:CONDition	Sets or queries the nibble data comparison condition of the SENT signal search fast channel.	5-224
:HEXa	Sets the SENT signal search fast channel nibble data in hexadecimal notation.	5-224

5.1 List of Commands

Command	Function	Page
:PATtern	Sets or queries the SENT signal search fast channel nibble data in binary notation.	5-224
:FSAComm?	Queries all fast channel status and communication nibble settings for SENT signal searching.	5-224
:FSAComm		
:HEXa	Sets the SENT signal search fast channel status and communication nibble data in hexadecimal notation.	5-224
:PATtern	Sets or queries the SENT signal search fast channel status and communication nibble data in binary notation.	5-224
:MODE	Sets or queries the SENT signal search mode.	5-225
:SDATa?	Queries all slow channel data settings for SENT signal searching.	5-225
:SDATa		
:ENHanced?	Queries all slow channel enhanced type settings for SENT signal searching.	5-225
:ENHanced		
:CBIT	Sets or queries the SENT signal search slow channel enhanced type configuration bit.	5-225
:D12Bit?	Queries all slow channel enhanced type 12-bit data and 8-bit ID settings for SENT signal searching.	5-225
:D12Bit		
:DATA?	Queries all slow channel enhanced type 12-bit data and 8-bit ID data settings for SENT signal searching.	5-225
:DATA		
:CONDition	Sets or queries the data comparison condition of slow channel enhanced type 12-bit data and 8-bit ID for SENT signal searching.	5-225
:DECimal<y>	Sets or queries the slow channel enhanced type 12-bit data and 8-bit ID data for SENT signal searching in decimal notation.	5-226
:HEXa	Sets the slow channel enhanced type 12-bit data and 8-bit ID data for SENT signal searching in hexadecimal notation.	5-226
:MODE	Sets or queries the data enable/disable condition of the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal searching.	5-226
:PATtern	Sets or queries the slow channel enhanced type 12-bit data and 8-bit ID data for SENT signal searching in binary notation.	5-226
:ID?	Queries all ID settings related to the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal searching.	5-226
:ID		
:CONDition	Sets or queries the ID comparison condition of slow channel enhanced type 12-bit data and 8-bit ID for SENT signal searching.	5-226
:DECimal<y>	Sets or queries the ID of the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal searching in decimal notation.	5-227
:MODE	Sets or queries the ID enable/disable condition of the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal searching.	5-227
:D16Bit?	Queries all slow channel enhanced type 16-bit data and 4-bit ID settings for SENT signal searching.	5-227
:D16Bit		
:DATA?	Queries all slow channel enhanced type 16-bit data and 4-bit ID data settings for SENT signal searching.	5-227
:DATA		
:CONDition	Sets or queries the data comparison condition of slow channel enhanced type 16-bit data and 4-bit ID for SENT signal searching.	5-227
:DECimal<y>	Sets or queries the slow channel enhanced type 16-bit data and 4-bit ID data for SENT signal searching in decimal notation.	5-227
:HEXa	Sets the slow channel enhanced type 16-bit data and 4-bit ID data for SENT signal searching in hexadecimal notation.	5-228
:MODE	Sets or queries the data enable/disable condition of the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal searching.	5-228
:PATtern	Sets or queries the slow channel enhanced type 16-bit data and 4-bit ID data for SENT signal searching in binary notation.	5-228
:ID?	Queries all ID settings related to the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal searching.	5-228
:ID		
:CONDition	Sets or queries the ID comparison condition of slow channel enhanced type 16-bit data and 4-bit ID for SENT signal searching.	5-228
:DECimal<y>	Sets or queries the ID of the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal searching in decimal notation.	5-228

5.1 List of Commands

Command	Function	Page
:MODE	Sets or queries the ID enable/disable condition of the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal searching.	5-229
:SHORT?	Queries all slow channel short type settings for SENT signal searching.	5-229
:SHORT		
:DATA?	Queries all slow channel short type data settings for SENT signal searching.	5-229
:DATA		
:CONDition	Sets or queries the short type data comparison condition of the SENT signal search slow channel.	5-229
:DECimal<y>	Sets or queries the SENT signal search slow channel short type data in decimal notation.	5-229
:HEXa	Sets the SENT signal search slow channel short type data in hexadecimal notation.	5-229
:MODE	Sets or queries the short type data enable/disable condition of the SENT signal search slow channel.	5-229
:PATtern	Sets or queries the SENT signal search slow channel short type data in binary notation.	5-230
:ID?	Queries all slow channel short type ID settings for SENT signal searching.	5-230
:ID		
:CONDition	Sets or queries the short type ID comparison condition of the SENT signal search slow channel.	5-230
:DECimal<y>	Sets or queries the SENT signal search slow channel short type ID in decimal notation.	5-230
:MODE	Sets or queries the short type ID enable/disable condition of the SENT signal search slow channel.	5-230
:TREND<y>?	Queries all SENT signal analysis trend display settings.	5-230
:TREND<y>		
:AScale	Executes auto scaling of the SENT signal analysis trend display.	5-230
:CURSor?	Queries all SENT signal analysis trend display cursor measurement settings.	5-231
:CURSor		
:DT		
:VALue?	Queries the time value between cursors on the SENT signal analysis trend display.	5-231
:DV	Queries the vertical value between cursors on the SENT signal analysis trend display.	5-231
:MODE	Sets or queries the automated measurement mode of the SENT signal analysis trend display.	5-231
:POSition<z>	Sets or queries the position of the specified cursor on the SENT signal analysis trend display.	5-231
:T<z>		
:VALue?	Queries the time value at the specified cursor on the SENT signal analysis trend display.	5-231
:V<z>	Queries the vertical value at the specified cursor on the SENT signal analysis trend display.	5-231
:DISPlay	Sets or queries whether to show (ON) or hide (OFF) the SENT signal analysis trend.	5-231
:FCONtrol	Sets or queries the frame control value of the SENT signal analysis trend display.	5-232
:HRANge	Sets or queries the SENT signal analysis trend display source window.	5-232
:SID	Sets or queries the slow channel ID of the SENT signal analysis trend display in hexadecimal notation.	5-232
:SOURce	Sets or queries the SENT signal analysis trend display source channel.	5-232
:UDATa	Sets or queries the fast channel user-defined data of the SENT signal analysis trend display.	5-232
:VERTical	Sets or queries the vertical range of the SENT signal analysis trend display.	5-232
:VTDisplay	Sets or queries the on/off status of the VT waveform display on the SENT signal analysis trend display.	5-233
:SOURce?	Queries all analysis and search settings.	5-233
:SOURce		
:{CHANnel<y> MATH<y>}?	Queries all source waveform settings.	5-233
:{CHANnel<y> MATH<y>}		
:HYSteresis	Sets or queries the source waveform hysteresis.	5-233
:LEVel	Sets or queries the source waveform level.	5-233
:SPATtern?	Queries all user-defined bus signal analysis and search settings.	5-233
:SPATtern		

5.1 List of Commands

Command	Function	Page
:ANALyze?	Queries all user-defined bus signal analysis execution settings.	5-233
[:ANALyze]		
:SETup?	Queries all user-defined bus signal analysis settings.	5-233
:SETup		
:BRATe	Sets or queries the user-defined bus signal analysis bit rate.	5-233
:CLOCK?	Queries all user-defined bus signal analysis clock signal settings.	5-234
:CLOCK		
:MODE	Sets or queries the enable/disable condition of the user-defined bus signal analysis clock signal.	5-234
:POLarity	Sets or queries the slope of the user-defined bus signal analysis clock signal.	5-234
:SOURce	Sets or queries the user-defined bus signal analysis clock signal.	5-234
:CS?	Queries all user-defined bus signal analysis chip select signal settings.	5-234
:CS		
:ACTive	Sets or queries the active state of the user-defined bus signal analysis chip select signal.	5-234
:SOURce	Sets or queries the user-defined bus signal analysis chip select signal.	5-235
:DATA?	Queries all user-defined bus signal analysis data signal settings.	5-235
:DATA		
:ACTive	Sets or queries the active state of the user-defined bus signal analysis data signal.	5-235
:SOURce	Sets or queries the user-defined bus signal analysis data signal.	5-235
:LATCh?	Queries all user-defined bus signal analysis latch signal settings.	5-235
:LATCh		
:POLarity	Sets or queries the slope of the user-defined bus signal analysis latch signal.	5-235
:SOURce	Sets or queries the user-defined bus signal analysis latch signal.	5-236
:SPoiNt	Sets or queries the analysis start point of the user-defined bus signal analysis.	5-236
:SEARch?	Queries all user-defined bus signal search settings.	5-236
:SEARch		
:ABORT	Aborts the user-defined bus signal search.	5-236
:EXECute	Executes a user-defined bus signal search.	5-236
:SElect	Sets which detected point to display in the user-defined bus signal search zoom window and queries the zoom position of the detected point.	5-236
:SElect? Maximum	Queries the number of detected points in the user-defined bus signal search.	5-236
:SETup?	Queries all User-defined bus signal search setup settings.	5-236
:SETup		
:BITSize	Sets or queries the user-defined bus signal search bit length.	5-236
:HEXa	Sets the data conditions of the user-defined bus signal searching in hexadecimal notation.	5-237
:PATtern	Sets or queries the data conditions of the user-defined bus signal searching in binary notation.	5-237
:SPI?	Queries all SPI bus signal analysis and search settings.	5-237
:SPI		
:ANALyze?	Queries all SPI bus signal analysis settings.	5-237
[:ANALyze]		
:SETup?	Queries all SPI bus signal analysis bus settings.	5-237
:SETup		
:BITorder	Sets or queries the bit order of the SPI bus signal analysis data.	5-237
:CLOCK?	Queries all SPI bus signal analysis clock signal settings.	5-237
:CLOCK		
:POLarity	Sets or queries the slope of the SPI bus signal analysis clock signal.	5-237
:SOURce	Sets or queries the SPI bus signal analysis clock signal.	5-238
:CS?	Queries all SPI bus signal analysis chip select signal settings.	5-238
:CS		
:ACTive	Sets or queries the active state of the SPI bus signal analysis chip select signal.	5-238
:SOURce	Sets or queries the SPI bus signal analysis chip select signal.	5-238
:DATA<y>?	Queries all SPI bus signal analysis data signal settings.	5-238
:DATA<y>		
:SOURce	Sets or queries the SPI bus signal analysis data signal.	5-238
:FIELD	Sets or queries the field size of the SPI bus signal analysis data.	5-239
:SPI		
[:ANALyze]		

5.1 List of Commands

Command	Function	Page
:SETup		
:GRouping	Sets or queries the grouping on/off state for when the SPI bus signal analysis chip select signal is not available.	5-239
:SERIalbus<x>		
:SPI		
[:ANALyze]		
:SETup		
:ITIME	Sets or queries the idle time for when the SPI bus signal analysis chip select signal is not available.	5-239
:MODE	Sets or queries the wiring system (three-wire system or four-wire system) of the SPI bus signal analysis data signal.	5-239
:MSBLsb	Sets or queries the MSB/LSB bit for SPI bus signal analysis data.	5-239
:TYPE	Sets or queries the SPI bus signal analysis data type.	5-239
:DETail?	Queries all SPI bus signal analysis result list settings.	5-239
:DETail		
:DISPlay	Sets or queries the display mode for the SPI bus signal analysis result list.	5-240
:LIST		
:ITEM?	Queries all items that will be displayed in the SPI bus signal analysis result list.	5-240
:VALue?	Queries all the data for the specified analysis number in the SPI bus signal analysis result list.	5-240
:SEARCh?	Queries all SPI bus signal search settings.	5-240
:SEARCh		
:ABORt	Aborts the SPI bus signal search.	5-240
:EXECute	Executes an SPI bus signal search.	5-240
:SElect	Sets which detected point to display in the SPI bus signal search zoom window and queries the zoom position of the detected point.	5-240
:SElect? MAXimum	Queries the number of detected points in the SPI bus signal search.	5-240
:SETup?	Queries all SPI bus signal search condition settings.	5-240
[:SETup]		
:DATA<y>?	Queries all SPI bus signal search data settings.	5-240
:DATA<y>		
:BCount	Sets or queries the pattern comparison start position of the specified data for SPI bus signal searching.	5-241
:CONDition	Sets or queries the judgment method (true/false) of the specified data for SPI bus signal searching.	5-241
:DBYTe	Sets or queries the data size (number of bytes) of the specified data for SPI bus signal searching.	5-241
:DPATtern?	Queries all SPI bus signal search data settings.	5-241
:DPATtern		
:HEXa	Sets the specified data for SPI bus signal searching in hexadecimal notation.	5-241
:PATtern	Sets or queries the specified data for SPI bus signal searching in binary notation.	5-241
:MODE	Sets or queries the data condition enable/disable state for SPI bus signal searching.	5-242
:TYPE	Sets or queries the search type.	5-242
:UART?	Queries all UART signal analysis and search settings.	5-242
:UART		
:ANALyze?	Queries all UART signal analysis settings.	5-242
[:ANALyze]		
:SETup?	Queries all bus setup settings for UART signal analysis.	5-242
:SETup		
:BITorder	Sets or queries the UART signal analysis bit order.	5-242
:BRATe	Sets or queries the UART signal analysis bit rate (data transfer rate).	5-242
:BSPace	Sets or queries the UART signal analysis byte space.	5-242
:GRouping	Sets or queries on/off state of UART signal analysis grouping.	5-243
:POLarity	Sets or queries the UART signal analysis polarity.	5-243
:SOURce	Sets or queries the UART signal analysis signal.	5-243
:SPOint	Sets or queries the UART signal analysis sample point.	5-243
:DETail?	Queries all UART signal analysis result list settings.	5-243
:DETail		
:DISPlay	Sets or queries the display mode for the UART signal analysis result list.	5-243
:LIST		

5.1 List of Commands

Command	Function	Page
:ITEM?	Queries all items that will be displayed in the UART signal analysis result list.	5-243
:VALue?	Queries all the data for the specified analysis number in the UART signal analysis result list.	5-244
:SEARCh?	Queries all UART signal search settings.	5-244
:SEARCh		
:ABORt	Aborts the UART signal search.	5-244
:EXECute	Executes a UART signal search.	5-244
:SELEct	Sets which detected point to display in the UART signal search zoom window and queries the zoom position of the detected point.	5-244
:SELEct? Maximum	Queries the number of detected points in the UART signal search.	5-244
:SETUp?	Queries all search conditions for UART signal searching.	5-244
:SETUp		
:DATA?	Queries all UART signal search data settings.	5-244
:DATA		
:ASCIi	Sets the UART signal search data in ASCII format.	5-244
:CONDition	Sets or queries the data judgment method (true/false) for UART signal searching.	5-244
:CSENSitive	Sets or queries whether to distinguish uppercase and lowercase letters in ASCII data for the UART signal search.	5-245
:DBYTe	Sets or queries the number of data bytes for UART signal searching.	5-245
:DPATtern?	Queries all UART signal search data settings.	5-245
:DPATtern		
:HEXa	Sets the data for UART signal searching in hexadecimal notation.	5-245
:PATtern	Sets or queries the data for UART signal searching in binary notation.	5-245
:PFORMat	Sets or queries the data input format, which is one of the ID and data conditions, for UART signal searching.	5-245
:ERRor?	Queries all UART signal search error settings.	5-245
:ERRor		
:FRAMing	Sets or queries the UART signal search framing error setting.	5-245
:PARity	Sets or queries the UART signal search parity error setting.	5-246
:PMODE	Sets or queries the UART signal search parity mode setting.	5-246
:FORMat	Sets or queries the UART signal search format setting.	5-246
:MODE	Sets or queries the UART signal search mode setting.	5-246
:ZLINKage	Sets or queries the linkage (on/off) between the result numbers and zoom positions of the serial bus signal analysis results.	5-246

SNAP Group

:SNAP	Takes a snapshot.	5-247

SSTart Group

:SSTart?	Sets the trigger mode to single, starts waveform acquisition. The instrument returns 0 if the waveform acquisition is stopped within the specified timeout. Otherwise, the instrument returns 1.	5-248

START Group

:START	Starts waveform acquisition.	5-249

STATus Group

:STATus?	Queries all the settings of the communication status feature.	5-250
:STATus		
:CONDition?	Queries the contents of the condition register.	5-250
:EESe	Sets or queries the extended event enable register.	5-250
:EESR?	Queries the contents of the extended event register and clears the register.	5-250
:ERRor?	Queries the error code and message of the last error that has occurred (top of the error queue).	5-250
:FILTer<x>	Sets or queries the transition filter.	5-250
:QENable	Sets or queries whether messages other than errors will be stored to the error queue.	5-250
:QMESsage	Sets or queries whether message information will be attached to the response to the STATus:ERRor? query.	5-250
:SPOLl?	Executes serial polling.	5-250

5.1 List of Commands

Command	Function	Page
STOP Group		
:STOP	Stops waveform acquisition.	5-251
STORe Group		
:STORe?	Queries all settings related to setup data in the internal memory.	5-252
:STORe		
:SETup<x>?	Queries all settings related to the setup data in the specified location of the internal memory.	5-252
:SETup<x>		
:COMMeNt	Sets or queries the comment for the setup data that is stored to the specified location in the internal memory.	5-252
:DATE?	Queries the date and time of the setup data that is stored to the specified location in the internal memory.	5-252
:EXECute	Saves setup data to the specified location in the internal memory.	5-252
:LOCK	Sets or queries the protection on/off state for the setup data stored to the specified location in the internal memory.	5-252
SYSTem Group		
:SYSTem?	Queries all system settings.	5-253
:SYSTem		
:BEEP	Generates a beep sound.	5-253
:CLICk	Sets or queries whether click sounds are produced.	5-253
:CLOCk?	Queries all date/time settings.	5-253
:CLOCk		
:DATE	Sets or queries the date.	5-253
:FORMat	Sets or queries the date format.	5-253
:MODE	Sets or queries whether the date and time are displayed.	5-253
:SNTP?	Queries the SNTP clock setting.	5-253
:SNTP		
:EXECute	Sets the clock using SNTP.	5-253
:GMTTime	Sets or queries the time difference from Greenwich Mean Time.	5-253
:TIME	Sets or queries the time.	5-253
:DCANcel	Sets or queries whether the specified delay is applied to time measurement values.	5-253
:DLMSync?	Queries all synchronous operation settings.	5-254
:DLMSync		
:ADESkew	Executes auto deskewing for synchronous operation.	5-254
:CONNeCt?	Starts synchronous operation and queries for the results.	5-254
:DESKew	Sets or queries the deskewing for synchronous operation.	5-254
:DISConnect?	Ends synchronous operation and queries for the results.	5-254
:LANGuage	Sets or queries the message language.	5-254
:LCD?	Queries all LCD settings.	5-254
:LCD		
:AUTO?	Queries all settings related to the LCD backlight auto-off feature.	5-254
:AUTO		
:MODE	Sets or queries on/off state of the LCD backlight auto-off feature.	5-254
:TIME	Sets or queries the LCD backlight auto-off time.	5-254
:BRIGhtness	Sets or queries the LCD brightness.	5-254
:MODE	Sets or queries the LCD backlight on/off state.	5-254
:MLANGuage	Sets or queries the menu language.	5-254
:MPARameter?	Queries all settings related to automated measurement values of waveform parameters and cursor measurement values.	5-255
:MPARameter		
:FSIZe	Sets or queries the font size that is used to display the automatically measured waveform parameters and cursor measurement values.	5-255
:MROWs	Sets or queries the number of rows of automatically measured waveform parameters and cursor measurement values to show from the menu display area.	5-255
:OCANcel	Sets or queries whether the specified offset voltage is applied to measurement results and computation results.	5-255
:POWeroff		

5.1 List of Commands

Command	Function	Page
:EXECute	Turns the instrument off. This command also sets how the instrument is to behave when the main power switch turned on the next time.	5-255
:PPOWer		
:STATus?	Queries the probe power supply status.	5-255
:TOUCh	Sets or queries the touch panel on/off state.	5-255
:TOUT?	Queries all trigger output settings.	5-255
:TOUT		
:POLarity	Sets or queries the trigger out polarity.	5-256
:RTIMing	Sets or queries the timing for resetting the trigger output.	5-256
:TSYNchro?	Queries all time synchronization settings.	5-256
:TSYNchro		
:IEEE1588?	Queries all IEEE 1588 time synchronization settings.	5-256
:IEEE1588		
:DNUMber	Sets or queries the domain number of IEEE 1588 time synchronization.	5-256
:LSECond	Sets or queries the leap second of IEEE 1588 time synchronization.	5-256
:NLAYer	Sets or queries the network layer of IEEE 1588 time synchronization.	5-256
:PRIority<x>	Sets or queries the priority of IEEE 1588 time synchronization.	5-256
:PState	Queries the state of IEEE 1588 time synchronization.	5-256
:SONLy	Sets or queries the Slave Only of IEEE 1588 time synchronization.	5-257
:State	Sets or queries the on/off status of IEEE 1588 time synchronization.	5-257
:STATe?	Queries the instrument's time synchronization status.	5-257
:USBKeyboard	Sets or queries the USB keyboard type.	5-257

TIMEbase Group

:TIMEbase?	Queries all time base settings.	5-258
:TIMEbase		
:SRATe?	Queries the sample rate.	5-258
:TDIV	Sets or queries the Time/div value.	5-258

TRIGger Group

:TRIGger?	Queries all trigger settings.	5-259
:TRIGger		
:ABN?	Queries all A->B(N) trigger settings.	5-259
:ABN		
:COUNT	Sets or queries the number of times condition B must be met for A->B(N) triggers.	5-259
:ACTioN?	Queries all action-on-trigger settings.	5-259
:ACTioN		
:ACQCount	Sets or queries the action count of the action-on-trigger.	5-259
:BUZZer	Sets or queries whether a warning sound is generated as an action.	5-260
:HCOpy	Sets or queries whether screen capture is output as an action.	5-260
:MAIL?	Queries all settings related to the mail notification on action.	5-260
:MAIL		
:COUNT	Sets or queries the upper limit to the number of mail notifications for the mail notification on action.	5-260
:MODE	Sets or queries whether a mail notification is made as an action.	5-260
:SAVE	Sets or queries whether waveform data is saved to the storage device as an action.	5-260
:START	Starts action-on-trigger.	5-260
:STOP	Stops action-on-trigger.	5-260
:ADB?	Queries all A Delay B trigger settings.	5-260
:ADB		
:DELay	Sets or queries the delay time for condition B for A Delay B triggers.	5-260
:{:ATRigger BTrigger}?	Queries all trigger condition settings.	5-260
{[:ATRigger] :BTrigger}		
:CAN?	Queries all CAN bus signal trigger settings.	5-260
:CAN		
:BRATe	Sets or queries the CAN bus signal trigger bit rate (data transfer rate).	5-261
:EFrame?	Queries all CAN bus signal trigger error settings.	5-261
:EFrame		
:CRC	Sets or queries the CAN bus signal trigger CRC error setting.	5-261
[:MODE]	Sets or queries the CAN bus signal trigger error frame setting.	5-261

5.1 List of Commands

Command	Function	Page
:STUFF	Sets or queries the CAN bus signal trigger stuff error setting.	5-261
:IDData?	Queries all ID and data condition settings for CAN bus signal triggering.	5-261
[:IDData]		
:ACK?	Queries the ACK setting, which is one of the ID and data conditions, for CAN bus signal triggering.	5-261
:ACK		
:MODE	Sets or queries the ACK mode, which is one of the ID and data conditions, for CAN bus signal triggering.	5-261
:TYPE	Sets or queries the ACK condition, which is one of the ID and data conditions, for CAN bus signal triggering.	5-262
:DATA?	Queries the data, which is one of the ID and data conditions, for CAN bus signal triggering.	5-262
:DATA		
:CONDition	Sets or queries the data comparison condition, which is one of the ID and data conditions, for CAN bus signal triggering.	5-262
:DECimal<x>	Sets a reference value, which is one of the ID and data conditions, for CAN bus signal triggering in decimal notation.	5-262
:DLC	Sets or queries the number of significant bytes (DLC), which is one of the ID and data conditions, for CAN bus signal triggering.	5-262
:DPATtern?	Queries all judgment data settings for the ID and data conditions for CAN bus signal triggering.	5-262
:DPATtern		
:HEXa	Sets a reference value, which is one of the ID and data conditions, for CAN bus signal triggering in hexadecimal notation.	5-262
:PATtern	Sets or queries the data, which is one of the ID and data conditions, for CAN bus signal triggering in binary notation.	5-263
:ENDian	Sets or queries the byte order of the reference values, which is one of the ID and data conditions, for CAN bus signal triggering.	5-263
:MODE	Sets or queries the data condition (enabled/disabled), which is one of the ID and data conditions, for CAN bus signal triggering.	5-263
:MSBLsb	Sets or queries the MSB and LSB bits for the data, which is one of the ID and data conditions, for CAN bus signal triggering.	5-263
:SIGN	Sets or queries the data sign, which is one of the ID and data conditions, for CAN bus signal triggering.	5-263
:IDENtifier?	Queries the identifier, which is one of the ID and data conditions, for CAN bus signal triggering.	5-263
:IDENtifier		
:ID?	Queries the ID, which is one of the ID and data conditions, for CAN bus signal triggering.	5-263
:ID		
:HEXa	Sets an ID, which is one of the ID and data conditions, for CAN bus signal triggering in hexadecimal notation.	5-264
:MODE	Sets or queries the ID condition (enabled/disabled), which is one of the ID and data conditions, for CAN bus signal triggering.	5-264
:PATtern	Sets or queries the ID condition, which is one of the ID and data conditions, for CAN bus signal triggering in binary notation.	5-264
:MFORmat	Sets or queries the ID frame format (standard or extended), which is one of the ID and data conditions, for CAN bus signal triggering.	5-264
:PFORmat	Sets or queries the ID input format, which is one of the ID and data conditions, for CAN bus signal triggering.	5-264
:MSIGnal?	Queries the message signal, which is one of the ID and data conditions, for CAN bus signal triggering.	5-264
:MSIGnal		
:MESSage		
:ITEM	Sets the message item, which is one of the ID and data conditions, for CAN bus signal triggering.	5-264
:SElect	Sets or queries the CAN bus signal trigger message signal setting.	5-265
:SIGnal?	Queries the signal, which is one of the ID and data conditions, for CAN bus signal triggering.	5-265
:SIGnal		
:CONDition	Sets or queries the signal data condition, which is one of the ID and data conditions, for CAN bus signal triggering.	5-265
:DECimal<x>	Sets a reference signal value, which is one of the ID and data conditions, for CAN bus signal triggering in decimal notation.	5-265

5.1 List of Commands

Command	Function	Page
:ITEM	Sets the signal item, which is one of the ID and data conditions, for CAN bus signal triggering.	5-265
:RTR	Sets or queries the RTR value, which is one of the ID and data conditions, for CAN bus signal triggering.	5-265
:IDOR?	Queries all ID OR condition settings for CAN bus signal triggering.	5-265
:IDOR		
:ACK?	Queries all ACK settings for the ID OR conditions for CAN bus signal triggering.	5-266
:ACK		
:MODE	Sets or queries the ACK mode, which is one of the ID OR conditions, for CAN bus signal triggering.	5-266
:TYPE	Sets or queries the ACK condition, which is one of the ID OR conditions, for CAN bus signal triggering.	5-266
:DATA?	Queries all data settings for the ID OR conditions for CAN bus signal triggering.	5-266
:DATA		
[:MODE]	Sets or queries the data enable/disable condition, which is one of the ID OR conditions, for CAN bus signal triggering.	5-266
:IDENTifier?	Queries all identifier settings for the ID OR conditions for CAN bus signal triggering.	5-266
:IDENTifier		
:ID<x>?	Queries all ID settings for the ID OR conditions for CAN bus signal triggering.	5-266
:ID<x>		
:HEXa	Sets the ID value, which is one of the ID OR conditions, for CAN bus signal triggering in hexadecimal notation.	5-266
:MODE	Sets or queries the ID enable/disable condition, which is one of the ID OR conditions, for CAN bus signal triggering.	5-267
:PATtern	Sets or queries the ID condition, which is one of the ID OR conditions, for CAN bus signal triggering in binary notation.	5-267
:MFOrmAt	Sets or queries the ID frame format (standard or extended), which is one of the ID OR conditions, for CAN bus signal triggering.	5-267
:MODE	Sets or queries the identifier enable/disable condition, which is one of the ID OR conditions, for CAN bus signal triggering.	5-267
:PFOrmAt	Sets or queries the ID input format, which is one of the ID OR conditions, for CAN bus signal triggering.	5-267
:MSIGnal<x>		
:MESSAge		
:ITEM	Sets the message item, which is one of the ID OR conditions, for CAN bus signal triggering.	5-267
:RTR	Sets or queries the RTR, which is one of the ID OR conditions, for CAN bus signal triggering.	5-267
:MODE	Sets or queries the CAN bus signal trigger mode.	5-268
:RECEssive	Sets or queries the CAN bus signal trigger recessive level.	5-268
:SOURce	Sets or queries the CAN bus signal trigger signal.	5-268
:SPOint	Sets or queries the CAN bus signal trigger sample point.	5-268
:CANFD?	Queries all CAN FD bus signal trigger settings.	5-268
:CANFD		
:BRATe	Sets or queries the CAN FD bus signal trigger bit rate (data transfer rate).	5-268
:DBRate	Sets or queries the CAN FD bus signal trigger data phase bit rate (data transfer rate).	5-268
:DSPoint	Sets or queries the CAN FD bus signal trigger data phase sample point.	5-268
:EFrame?	Queries all CAN FD bus signal trigger error settings.	5-269
:EFrame		
:CRC	Sets or queries the CAN FD bus signal trigger CRC error setting.	5-269
:CRCEFactor?	Queries all CRC error factor settings for CAN FD bus signal triggering.	5-269
:CRCEFactor		
:CRCSequence	Sets or queries the CRC sequence (a CRC error factor) for CAN FD bus signal triggering.	5-269
:SCount	Sets or queries the Stuff Count (a CRC error factor) for CAN FD bus signal triggering.	5-269
:FSTuff	Sets or queries the CAN FD bus signal trigger fixed stuff error setting.	5-269
[:MODE]	Sets or queries the CAN FD bus signal trigger error frame setting.	5-269
:STUFF	Sets or queries the CAN FD bus signal trigger stuff error setting.	5-270

5.1 List of Commands

Command	Function	Page
:FDF?	Queries all CAN FD bus signal trigger FDF settings.	5-270
:FDF		
:CONDition	Sets or queries the CAN FD bus signal trigger FDF.	5-270
:FDStandard	Sets or queries whether the CAN FD bus signal for triggering is an ISO standard signal.	5-270
:IDData?	Queries all ID and data condition settings for CAN FD bus signal triggering.	5-270
[:IDData]		
:ACK?	Queries all ACK settings for the ID and data conditions for CAN FD bus signal triggering.	5-270
:ACK		
:MODE	Sets or queries the ACK mode, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-270
:TYPE	Sets or queries the ACK condition, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-270
:DATA?	Queries all data settings for the ID and data conditions for CAN FD bus signal triggering.	5-270
:DATA		
:BCount	Sets or queries the pattern comparison position, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-271
:CONDition	Sets or queries the comparison condition, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-271
:DBYTe	Sets or queries the number of data bytes, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-271
:DECimal<x>	Sets or queries a reference values (a or b), which is one of the ID and data conditions, for CAN FD bus signal triggering in decimal notation.	5-271
:DPATtern?	Queries all judgment data settings for the ID and data conditions for CAN FD bus signal triggering.	5-271
:DPATtern		
:HEXa	Sets a reference value, which is one of the ID and data conditions, for CAN FD bus signal triggering in hexadecimal notation.	5-271
:PATtern	Sets or queries the data value, which is one of the ID and data conditions, for CAN FD bus signal triggering in binary notation.	5-272
:ENDian	Sets or queries the byte order of the reference values, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-272
:MODE	Sets or queries the data enable/disable condition, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-272
:MSBLsb	Sets or queries the MSB and LSB bits for the data, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-272
:SIGN	Sets or queries whether signed or unsigned data format will be used, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-272
:IDENtifier?	Queries all identifier settings for the ID and data conditions for CAN FD bus signal triggering.	5-272
:IDENtifier		
:ID?	Queries all identifier settings for the ID and data conditions for CAN FD bus signal triggering.	5-272
:ID		
:HEXa	Sets the ID value, which is one of the ID and data conditions, for CAN FD bus signal triggering in hexadecimal notation.	5-273
:MODE	Sets or queries the ID enable/disable condition, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-273
:PATtern	Sets or queries the ID condition, which is one of the ID and data conditions, for CAN FD bus signal triggering in binary notation.	5-273
:MFORmat	Sets or queries the ID frame format (standard or extended), which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-273
:PFORmat	Sets or queries the ID input format, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-273
:MSIGnal?	Queries all message and signal settings for the ID and data conditions for CAN FD bus signal triggering.	5-273
:MSIGnal		
:MESSage		
:ITEM	Sets the message item, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-273
:SElect	Sets or queries the message and signal condition, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-274

5.1 List of Commands

Command	Function	Page
:SIGNal?	Queries all signal settings for the ID and data conditions for CAN FD bus signal triggering.	5-274
:SIGNal		
:CONDition	Sets or queries the signal condition, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-274
:DECimal<x>	Sets or queries a signal's reference value, which is one of the ID and data conditions, for CAN FD bus signal triggering in decimal notation.	5-274
:ITEM	Sets the signal item, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-274
:RTR	Sets or queries the RTR, which is one of the ID and data conditions, for CAN FD bus signal triggering.	5-274
:IDOR?	Queries all ID OR condition settings for CAN FD bus signal triggering.	5-274
:IDOR		
:ACK?	Queries all ACK settings for the ID OR conditions for CAN FD bus signal triggering.	5-275
:ACK		
:MODE	Sets or queries the ACK mode, which is one of the ID OR conditions, for CAN FD bus signal triggering.	5-275
:TYPE	Sets or queries the ACK condition, which is one of the ID OR conditions, for CAN FD bus signal triggering.	5-275
:DATA?	Queries all data settings for the ID OR conditions for CAN FD bus signal triggering.	5-275
:DATA		
[:MODE]	Sets or queries the data enable/disable condition, which is one of the ID OR conditions, for CAN FD bus signal triggering.	5-275
:IDENTifier?	Queries all identifier settings for the ID OR conditions for CAN FD bus signal triggering.	5-275
:IDENTifier		
:ID<x>?	Queries all ID settings for the ID OR conditions for CAN FD bus signal triggering.	5-275
:ID<x>		
:HEXa	Sets the ID value, which is one of the ID OR conditions, for CAN FD bus signal triggering in hexadecimal notation.	5-275
:MODE	Sets or queries the ID enable/disable condition, which is one of the ID OR conditions, for CAN FD bus signal triggering.	5-276
:PATtern	Sets or queries the ID condition, which is one of the ID OR conditions, for CAN FD bus signal triggering in binary notation.	5-276
:MForma	Sets or queries the ID frame format (standard or extended), which is one of the ID OR conditions, for CAN FD bus signal triggering.	5-276
:MODE	Sets or queries the ID enable/disable condition, which is one of the ID OR conditions, for CAN FD bus signal triggering.	5-276
:PForma	Sets or queries the ID input format, which is one of the ID OR conditions, for CAN FD bus signal triggering.	5-276
:MSIGNal<x>		
:MESSa		
:ITEM	Sets the message item, which is one of the ID OR conditions, for CAN FD bus signal triggering.	5-276
:RTR	Sets or queries the RTR, which is one of the ID OR conditions, for CAN FD bus signal triggering.	5-276
:MODE	Sets or queries the CAN FD bus signal trigger mode.	5-277
:RECEssive	Sets or queries the CAN FD bus signal trigger recessive level.	5-277
:SOURce	Sets or queries the CAN FD bus signal trigger signal.	5-277
:SPOint	Sets or queries the CAN FD bus signal trigger sample point.	5-277
:CXPI?	Queries all CXPI bus signal trigger settings.	5-277
:CXPI		
:BRATe	Sets or queries the CXPI bus signal trigger bit rate (data transfer rate).	5-277
:CTOLerance	Sets or queries the CXPI bus signal trigger clock tolerance.	5-277
:ERRor?	Queries all CXPI bus signal trigger error settings.	5-277
:ERRor		
:CLOCK	Sets or queries the CXPI bus signal trigger clock error setting.	5-278
:CRC	Sets or queries the CXPI bus signal trigger CRC error setting.	5-278
:DLENgth	Sets or queries the CXPI bus signal trigger data length error setting.	5-278
:FRAMing	Sets or queries the CXPI bus signal trigger framing error setting.	5-278
:IBS	Sets or queries the CXPI bus signal trigger IBS error setting.	5-278

5.1 List of Commands

Command	Function	Page
:PARity	Sets or queries the CXPI bus signal trigger parity error setting.	5-278
:IDData?	Queries all ID and data condition settings for CXPI bus signal triggering.	5-278
:IDData		
:DATA?	Queries the data, which is one of the ID and data conditions, for CXPI bus signal triggering.	5-279
:DATA		
:BCOunt	Sets or queries the pattern comparison position, which is one of the ID and data conditions, for CXPI bus signal triggering.	5-279
:CONDItion	Sets or queries the data comparison condition, which is one of the ID and data conditions, for CXPI bus signal triggering.	5-279
:DBYTe	Sets or queries the number of data bytes, which is one of the ID and data conditions, for CXPI bus signal triggering.	5-279
:DECimal<x>	Sets a reference value, which is one of the ID and data conditions, for CXPI bus signal triggering in decimal notation.	5-279
:DPATtern?	Queries the ID, which is one of the ID and data conditions, for CXPI bus signal triggering.	5-279
:DPATtern		
:HEXa	Sets an ID, which is one of the ID and data conditions, for CXPI bus signal triggering in hexadecimal notation.	5-279
:PATtern	Sets or queries the ID condition, which is one of the ID and data conditions, for CXPI bus signal triggering in binary notation.	5-280
:ENDian	Sets or queries the byte order of the reference values, which is one of the ID and data conditions, for CXPI bus signal triggering.	5-280
:MODE	Sets or queries the ID and data condition (enabled/disabled) for CXPI bus signal triggering.	5-280
:MSBLsb	Sets or queries the MSB and LSB bits for the data, which is one of the ID and data conditions, for CXPI bus signal triggering.	5-280
:SIGN	Sets or queries the data sign, which is one of the ID and data conditions, for CXPI bus signal triggering.	5-280
:FINformation	Queries all frame information settings for the ID and data conditions for CXPI bus signal triggering.	5-280
:CT	Sets or queries the frame information counter value, which is one of the ID and data conditions, for CXPI bus signal triggering.	5-280
:MODE	Sets or queries the frame information (enabled/disabled), which is one of the ID and data conditions, for CXPI bus signal triggering.	5-281
:SLEEP	Sets or queries the frame information sleep bit, which is one of the ID and data conditions, for CXPI bus signal triggering.	5-281
:WAKEup	Sets or queries the frame information wakeup bit, which is one of the ID and data conditions, for CXPI bus signal triggering.	5-281
:ID?	Queries the ID, which is one of the ID and data conditions, for CXPI bus signal triggering.	5-281
:ID		
:HEXa	Sets an ID, which is one of the ID and data conditions, for CXPI bus signal triggering in hexadecimal notation.	5-281
:MODE	Sets or queries the ID condition (enabled/disabled), which is one of the ID and data conditions, for CXPI bus signal triggering.	5-281
:PATtern	Sets or queries the ID, which is one of the ID and data conditions, for CXPI bus signal triggering in binary notation.	5-281
:PTYPE	Sets or queries the PTYPE comparison condition, which is one of the ID and data conditions, for CXPI bus signal triggering.	5-282
:IDOR?	Queries all ID OR condition settings for CXPI bus signal triggering.	5-282
:IDOR		
:IDENtifier?	Queries all identifier settings for the IDOR conditions for CXPI bus signal triggering.	5-282
:IDENtifier		
:ID<x>?	Queries all ID settings for the IDOR conditions for CXPI bus signal triggering.	5-282
:ID<x>		
:HEXa	Sets the ID value, which is one of the IDOR conditions, for CXPI bus signal triggering in hexadecimal notation.	5-282
:MODE	Sets or queries the ID enable/disable condition, which is one of the IDOR conditions, for CXPI bus signal triggering.	5-282
:PATtern	Sets or queries the ID condition, which is one of the IDOR conditions, for CXPI bus signal triggering in binary notation.	5-282

5.1 List of Commands

Command	Function	Page
:MODE	Sets or queries the identifier enable/disable condition, which is one of the IDOR conditions, for CXPI bus signal triggering.	5-282
:PTYPE	Sets or queries the comparison condition of PTYPE, which is an IDOR condition, for CXPI bus signal triggering.	5-283
:MODE	Sets or queries the CXPI bus signal trigger type.	5-283
:PTYPE?	Queries all CXPI bus signal trigger PTYPE settings.	5-283
:PTYPE		
:CONdition	Sets or queries the CXPI bus signal trigger PTYPE condition setting.	5-283
:SOURce	Sets or queries the CXPI bus signal trigger signal.	5-283
:TSAmple	Sets or queries the logic value (1 or 0) determination threshold for CXPI bus signal triggering.	5-283
:WAKEupsleep	Queries all wakeup and sleep settings for CXPI bus signal triggering.	5-283
:TYPE	Sets or queries the wakeup and sleep conditions for CXPI bus signal triggering.	5-283
:FLEXray?	Queries all FlexRay bus signal trigger settings.	5-284
:FLEXray		
:BCHannel	Sets or queries the channel bus type for FlexRay bus signal triggering.	5-284
:BRATe	Sets or queries the FlexRay bus signal trigger bit rate (data transfer rate).	5-284
:ERRor?	Queries all FlexRay bus signal trigger error settings.	5-284
:ERRor		
:BSS	Sets or queries the FlexRay bus signal trigger BSS error setting.	5-284
:CRC	Sets or queries the FlexRay bus signal trigger CRC error setting.	5-284
:FES	Sets or queries the FlexRay bus signal trigger FES error setting.	5-284
:IDData?	Queries all ID and data condition settings for FlexRay bus signal triggering.	5-284
[:IDData]		
:CCOunt?	Queries the cycle count, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-285
:CCOunt		
:CONdition	Sets or queries the cycle count data condition, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-285
:COUnT<x>	Sets or queries the cycle count, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-285
:MODE	Sets or queries the cycle count condition (enabled/disabled), which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-285
:DATA<x>?	Queries the data field, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-286
:DATA<x>		
:BCOunt	Sets or queries the pattern comparison position of the data field data, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-286
:CONdition	Sets or queries the data field data condition, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-286
:DBYTe	Sets or queries the number of bytes of the data field, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-286
:DECimal<y>	Sets a reference value, which is one of the ID and data conditions, for FlexRay bus signal triggering in hexadecimal notation.	5-286
:DPATtern?	Queries all data field judgment data settings for the ID and data conditions for FlexRay bus signal triggering.	5-286
:DPATtern		
:HEXa	Sets a reference value for the data field, which is one of the ID and data conditions, for FlexRay bus signal triggering in hexadecimal notation.	5-287
:PATtern	Sets or queries the data field data, which is one of the ID and data conditions, for FlexRay bus signal triggering in binary notation.	5-287
:ENDian	Sets or queries the data field endian, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-287
:MODE	Sets or queries the data field condition (enabled/disabled), which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-287
:MSBLsb	Sets or queries the MSB/LSB bit of the data field data, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-287
:SIGN	Sets or queries the data sign of the data field, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-287
:FID?	Queries the frame ID, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-288
:FID		

5.1 List of Commands

Command	Function	Page
:CONDition	Sets or queries the frame ID data condition, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-288
:ID<x>	Sets or queries the frame ID value, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-288
:MODE	Sets or queries the frame ID condition (enabled/disabled), which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-288
:INDicator?	Queries the indicator, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-288
:INDicator		
:MODE	Sets or queries the indicator condition (enabled/disabled), which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-288
:NFRame	Sets or queries the indicator null frame, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-289
:PPReamble	Sets or queries the indicator payload preamble, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-289
:STFRame	Sets or queries the indicator start frame, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-289
:SYFRame	Sets or queries the indicator sync frame, which is one of the ID and data conditions, for FlexRay bus signal triggering.	5-289
:IDOR?	Queries all OR condition settings for FlexRay bus signal triggering.	5-289
:IDOR		
:ID<x>?	Queries all settings related to the specified frame ID of the OR condition for FlexRay bus signal triggering.	5-289
:ID<x>		
:CCount?	Queries all settings related to the cycle count of the specified frame ID of the OR condition for FlexRay bus signal triggering.	5-289
:CCount		
:CONDition	Sets or queries the cycle count of the specified frame ID of the OR condition for FlexRay bus signal triggering.	5-290
:COUNT<y>	Sets or queries the reference cycle count value of the specified frame ID of the OR condition for FlexRay bus signal triggering.	5-290
:FID?	Queries all settings related to the ID condition of the specified frame ID of the OR condition for FlexRay bus signal triggering.	5-290
:FID		
:CONDition	Sets or queries the ID judgment condition of the specified frame ID of the OR condition for FlexRay bus signal triggering.	5-290
:ID<y>	Sets or queries the ID reference value of the specified frame ID of the OR condition for FlexRay bus signal triggering.	5-290
:MODE	Sets or queries the specified frame ID condition (enabled/disabled) of the OR condition for FlexRay bus signal triggering.	5-290
:MODE	Sets or queries the FlexRay bus signal trigger mode.	5-291
:SOURce	Sets or queries the FlexRay bus signal trigger source.	5-291
:I2C?	Queries all I2C bus signal trigger settings.	5-291
:I2C		
:ADDRess?	Queries all I2C bus signal trigger address pattern settings.	5-291
:ADDRess		
:ADDRess?	Queries all settings related to the address of the I2C bus signal trigger address pattern.	5-291
:ADDRess		
:BIT10Address?	Queries all I2C bus signal trigger 10-bit address settings.	5-291
:BIT10Address		
:DIRection	Sets or queries the 10-bit address data direction for I2C bus signal triggering.	5-291
:HEXa	Sets the 10-bit address for I2C bus signal triggering in hexadecimal notation.	5-291
:PATtern	Sets or queries the 10-bit address for I2C bus signal triggering in binary notation.	5-291
:BIT7Address?	Queries all I2C bus signal trigger 7-bit address settings.	5-292
[:BIT7Address]		
:DIRection	Sets or queries the 7-bit address data direction for I2C bus signal triggering.	5-292
:HEXa	Sets the 7-bit address for I2C bus signal triggering in hexadecimal notation.	5-292
:PATtern	Sets or queries the 7-bit address for I2C bus signal triggering in binary notation.	5-292
:BIT7APsub?	Queries all I2C bus signal trigger 7-bit + sub address settings.	5-292
:BIT7APsub		

5.1 List of Commands

Command	Function	Page
:ADDRESS?	Queries all settings related to the 7-bit address of the 7-bit + sub address for I2C bus signal triggering.	5-292
:ADDRESS		
:DIRection	Sets or queries the 7-bit + sub address data direction for I2C bus signal triggering.	5-292
:HEXa	Sets the 7-bit address of 7-bit + sub address for I2C bus signal triggering in hexadecimal notation.	5-292
:PATtern	Sets or queries the 7-bit address of 7-bit + sub address for I2C bus signal triggering in binary notation.	5-293
:SADDRESS?	Queries all settings related to the sub address of the 7-bit + sub address for I2C bus signal triggering.	5-293
:SADDRESS		
:HEXa	Sets the sub address of 7-bit + sub address for I2C bus signal triggering in hexadecimal notation.	5-293
:PATtern	Sets or queries the sub address of 7-bit + sub address for I2C bus signal triggering in binary notation.	5-293
:MODE	Sets or queries the address condition enable/disable state for I2C bus signal triggering.	5-293
:TYPE	Sets or queries the address condition address format for I2C bus signal triggering.	5-293
:DATA?	Queries all settings related to the data of the I2C bus signal trigger address pattern.	5-294
:DATA		
:BCount	Sets or queries the data pattern comparison position for I2C bus signal triggering.	5-294
:BMode	Sets or queries on/off state of the data pattern comparison position for I2C bus signal triggering.	5-294
:CONDition	Sets or queries the data judgment method (true/false) for I2C bus signal triggering.	5-294
:DBYte	Sets or queries the number of setup data values for I2C bus signal triggering.	5-294
:DPATtern?	Queries all I2C bus signal trigger data settings.	5-294
:DPATtern		
:HEXa	Sets the data for I2C bus signal triggering in hexadecimal notation.	5-294
:PATtern	Sets or queries the data condition data for I2C bus signal triggering in binary notation.	5-294
:MODE	Sets or queries the data condition enable/disable state for I2C bus signal triggering.	5-295
:GENeralcall?	Queries all I2C bus signal trigger general call settings.	5-295
:GENeralcall		
:BIT7Maddress?	Queries all settings related to the 7-bit master address of general call for I2C bus signal triggering.	5-295
:BIT7Maddress		
:HEXa	Sets the 7-bit master address of general call for I2C bus signal triggering in hexadecimal notation.	5-295
:PATtern	Sets or queries the 7-bit master address of general call for I2C bus signal triggering in binary notation.	5-295
:SBYte	Sets or queries the general call second byte type for I2C bus signal triggering.	5-295
:INCLuderw	Sets or queries the enable/disable condition of address R/W for I2C bus signal triggering.	5-295
:MODE	Sets or queries the I2C bus signal trigger mode.	5-296
:NONack?	Queries all I2C bus signal trigger NON ACK ignore mode settings.	5-296
:NONack		
:HSMODE	Sets or queries whether NONACK is to be ignored in high-speed mode of I2C bus signal triggering.	5-296
:READaccess	Sets or queries whether NON ACK is to be ignored in the read access byte of I2C bus signal triggering.	5-296
:STARtbyte	Sets or queries whether NON ACK is to be ignored in the start byte of I2C bus signal triggering.	5-296
:SCL	Sets or queries the I2C bus signal trigger clock signal.	5-296
:SDA	Sets or queries the I2C bus signal trigger data signal.	5-296
:LIN?	Queries all LIN bus signal trigger settings.	5-296
:LIN		
:BLENgth	Sets or queries the LIN bus signal trigger break length.	5-297

5.1 List of Commands

Command	Function	Page
:BRATe	Sets or queries the LIN bus signal trigger bit rate (data transfer rate).	5-297
:ERRor?	Queries all LIN bus signal trigger error settings.	5-297
:ERRor		
:PARity	Sets or queries the LIN bus signal trigger parity error setting.	5-297
:SYNCh	Sets or queries the LIN bus signal trigger synch error setting.	5-297
:IDData?	Queries all ID and data settings for LIN bus signal triggering.	5-297
:IDData		
:DATA?	Queries all LIN bus signal trigger data settings.	5-297
:DATA		
:CONDition	Sets or queries the LIN bus signal trigger data condition setting.	5-297
:DBYTe	Sets or queries the number of data bytes for LIN bus signal triggering.	5-298
:DECimal<x>	Sets or queries the data for LIN bus signal triggering in decimal notation.	5-298
:DPATtern?	Queries all LIN bus signal trigger data settings.	5-298
:DPATtern		
:HEXa	Sets the data for LIN bus signal triggering in hexadecimal notation.	5-298
:PATtern	Sets or queries the data for LIN bus signal triggering in binary notation.	5-298
:ENDian	Sets or queries the data endian for LIN bus signal triggering.	5-298
:MODE	Sets or queries the data condition enable/disable state for LIN bus signal triggering.	5-298
:MSBLsb	Sets or queries the MSB/LSB bit for LIN bus signal triggering.	5-299
:SIGN	Sets or queries the data sign for LIN bus signal triggering.	5-299
:IDENtifier?	Queries all LIN bus signal trigger identifier settings.	5-299
:IDENtifier		
:ID?	Queries all LIN bus signal trigger ID settings.	5-299
:ID		
:HEXa	Sets the ID for LIN bus signal triggering in hexadecimal notation.	5-299
:MODE	Sets or queries the ID condition (enabled/disabled), which is one of the ID and data conditions, for LIN bus signal triggering.	5-299
:PATtern	Sets or queries the ID for LIN bus signal triggering in binary notation.	5-299
:IDOR?	Queries all ID OR condition settings for LIN bus signal triggering.	5-299
:IDOR		
:IDENtifier?	Queries all ID settings for the ID OR conditions for LIN bus signal triggering.	5-299
:IDENtifier		
:ID<x>?	Queries all settings related to the specified ID of the ID OR condition for LIN bus signal triggering.	5-299
:ID<x>		
:HEXa	Sets the ID value, which is one of the ID OR conditions, for LIN bus signal triggering in hexadecimal notation.	5-300
:MODE	Sets or queries the ID enable/disable condition, which is one of the ID OR conditions, for LIN bus signal triggering.	5-300
:PATtern	Sets or queries the ID condition, which is one of the ID OR conditions, for LIN bus signal triggering in binary notation.	5-300
:MODE	Sets or queries the ID condition enable/disable state for LIN bus signal triggering.	5-300
:MODE	Sets or queries the LIN bus signal trigger mode.	5-300
:SOURce	Sets or queries the LIN bus signal trigger signal.	5-300
:SPOint	Sets or queries the LIN bus signal trigger sample point.	5-300
:OR?	Queries all edge OR trigger settings.	5-301
:OR		
:ALL	Sets or queries whether to make all channels edge OR trigger sources at once.	5-301
:CHANnel<x>	Sets or queries the slope of each channel of the edge OR trigger.	5-301
:PATtern?	Queries all settings related to the combination (pattern) trigger of multiple inputs.	5-301
:PATtern		
:BITS?	Queries all settings related to the logic inputs of the combination (pattern) trigger of multiple inputs.	5-301
:BITS		
:HEXa	Sets the logic input state of the combination (pattern) trigger of multiple inputs in hexadecimal notation.	5-301
:PATtern	Sets or queries the logic input pattern of the combination (pattern) trigger of multiple inputs.	5-301

5.1 List of Commands

Command	Function	Page
:CHANnel<x>	Sets or queries the pattern of the specified channel for the combination (pattern) trigger of multiple inputs.	5-301
:CLOCK?	Queries all settings related to the clock signal for the combination (pattern) trigger of multiple inputs.	5-301
:CLOCK		
:SLOPe	Sets or queries the clock signal slope for the combination (pattern) trigger of multiple inputs.	5-302
:SOURce	Sets or queries the clock signal for the combination (pattern) trigger of multiple inputs.	5-302
:CONDition	Sets or queries the trigger condition for the combination (pattern) trigger of multiple inputs.	5-302
:LOGic	Sets or queries the trigger combination condition for the combination (pattern) trigger of multiple inputs.	5-302
:TQQualify	Sets or queries the time condition for combination (pattern) triggering of multiple inputs.	5-302
:PSI5?	Queries all PSI5 signal trigger settings.	5-302
:PSI5		
:DATA?	Queries all PSI5 signal data trigger settings.	5-302
:DATA		
:CONDition	Sets or queries the comparison condition of the PSI5 signal data trigger.	5-303
:DECimal<x>	Sets or queries the data of the PSI5 signal data trigger in decimal notation.	5-303
:DPATtern?	Queries all data settings of the PSI5 signal data triggering.	5-303
:DPATtern		
:HEXa	Sets the data of the PSI5 signal data trigger in hexadecimal notation.	5-303
:PATtern	Sets or queries the data of the PSI5 signal data trigger in binary notation.	5-303
:FRAMEinslot?	Queries all slot specification settings of the PSI5 signal data trigger.	5-303
:FRAMEinslot		
:MODE	Sets or queries the slot specification condition (enabled/disabled) of the PSI5 signal data trigger.	5-303
:SNUMber	Sets or queries the slot number of the PSI5 signal data trigger.	5-304
:PFORMat	Sets or queries the data input format of the PSI5 signal data trigger in binary notation.	5-304
:ERRor?	Queries all PSI5 signal error trigger settings.	5-304
:ERRor		
:FNUMber	Sets or queries the PSI5 signal frame number error trigger.	5-304
:FRAME	Sets or queries the PSI5 signal frame error trigger.	5-304
:MANChester	Sets or queries the PSI5 signal Manchester Error trigger.	5-304
:PCRC	Sets or queries the PSI5 signal parity/CRC error trigger.	5-304
:SBIT	Sets or queries the PSI5 signal start bit error trigger.	5-304
:SBOUNDary	Sets or queries the PSI5 signal slot boundary error trigger.	5-305
:FRAMEinslot?	Queries all PSI5 signal slot specification trigger settings.	5-305
:FRAMEinslot		
:SNUMber	Sets or queries the slot number of the PSI5 signal slot specification trigger.	5-305
:MODE	Sets or queries the PSI5 signal trigger mode.	5-305
:SETup?	Queries all bus setup settings for PSI5 signal triggering.	5-305
:SETup		
:DATA?	Queries all PSI5 signal trigger data signal settings.	5-305
:DATA		
:BRATe	Sets or queries the PSI5 signal trigger bit rate (data transfer rate).	5-305
:CTOLerance	Sets or queries the PSI5 signal trigger clock tolerance range.	5-305
:DBITs	Sets or queries the data length for PSI5 signal triggering.	5-306
:EDETection	Sets or queries the error detection method for PSI5 signal triggering.	5-306
:SNRejection?	Queries all PSI5 signal trigger sync noise rejection settings.	5-306
:SNRejection		
:ETIme	Sets or queries the rejection end time for PSI5 signal trigger sync noise rejection.	5-306
:MODE	Sets or queries the on/off state of PSI5 signal trigger sync noise rejection.	5-306
:SOURce	Sets or queries the data source for PSI5 signal triggering.	5-306
:NUMBerofslot	Sets or queries the number of slots for PSI5 signal trigger.	5-307
:SLOT<x>	Sets or queries the slot's start and end times for PSI5 signal trigger.	5-307
:SYNC	Sets or queries the sync signal for PSI5 signal triggering.	5-307
:SENT?	Queries all SENT signal trigger settings.	5-307
:SENT		

5.1 List of Commands

Command	Function	Page
:CEFactor?	Queries all SENT signal trigger error factor settings.	5-307
:CEFactor		
:SAComm?	Queries all status and communication error factor settings of SENT signal triggering.	5-307
:SAComm		
:BIT<x>	Sets or queries status and communication error factor bit of SENT signal triggering.	5-307
:SCPulses	Sets or queries the consecutive calibration pulse error factor of SENT signal triggering.	5-307
:CRCType	Sets or queries the SENT signal trigger CRC computation type.	5-308
:CTICk	Sets or queries the SENT signal trigger clock tick value.	5-308
:CTOLerance	Sets or queries the SENT signal trigger clock tolerance.	5-308
:DNIBbles	Sets or queries the number of data nibbles for SENT signal triggering.	5-308
:DTYPe	Sets or queries the SENT signal trigger fast channel data format.	5-308
:ERRor?	Queries all SENT signal trigger error settings.	5-308
:ERRor		
:FCRC	Sets or queries the SENT signal trigger fast channel CRC error.	5-308
:NDValue	Sets or queries the SENT signal trigger nibble data value error.	5-309
:NNUmber	Sets or queries the SENT signal trigger nibble data count error.	5-309
:SAComm	Sets or queries the SENT signal trigger status and communication error.	5-309
:SCPulses	Sets or queries the consecutive calibration pulse error for SENT signal triggering.	5-309
:SCRC	Sets or queries the SENT signal trigger slow channel CRC error.	5-309
:FDATa?	Queries all fast channel data settings for SENT signal triggering.	5-309
:FDATa		
:DATA<x>?	Queries all settings related to the specified fast channel user data for SENT signal triggering.	5-309
:DATA<x>		
:CONDition	Sets or queries the comparison condition of the specified user data of the SENT signal trigger fast channel.	5-310
:DECimal<y>	Sets or queries the data of the specified user data of the SENT signal trigger fast channel in decimal notation.	5-310
:MODE	Sets or queries the data enable/disable condition of the specified user data of the SENT signal trigger fast channel.	5-310
:DNIBbles?	Queries all fast channel nibble data settings for SENT signal triggering.	5-310
:DNIBbles		
:CONDition	Sets or queries the nibble data comparison condition of the SENT signal trigger fast channel.	5-310
:HEXa	Sets the SENT signal trigger fast channel nibble data in hexadecimal notation.	5-310
:PATtern	Sets or queries the SENT signal trigger fast channel nibble data in binary notation.	5-310
:FSAComm?	Queries all fast channel status and communication nibble settings for SENT signal triggering.	5-311
:FSAComm		
:HEXa	Sets the SENT signal trigger fast channel status and communication nibble data in hexadecimal notation.	5-311
:PATtern	Sets or queries the SENT signal trigger fast channel status and communication nibble data in binary notation.	5-311
:MODE	Sets or queries the SENT signal trigger mode.	5-311
:PPULse	Sets or queries the presence or absence of pause pulses for SENT signal triggering.	5-311
:SDATa?	Queries all slow channel data settings for SENT signal triggering.	5-311
:SDATa		
:ENHanced?	Queries all slow channel enhanced type settings for SENT signal triggering.	5-311
:ENHanced		
:CBIT	Sets or queries the SENT signal trigger slow channel enhanced type configuration bit.	5-311
:D12Bit?	Queries all slow channel enhanced type 12-bit data and 8-bit ID settings for SENT signal triggering.	5-311
:D12Bit		
:DATA?	Queries all slow channel enhanced type 12-bit data and 8-bit ID data settings for SENT signal triggering.	5-312

5.1 List of Commands

Command	Function	Page
:DATA		
:CONDition	Sets or queries the data comparison condition of slow channel enhanced type 12-bit data and 8-bit ID for SENT signal triggering.	5-312
:DECimal<x>	Sets or queries the slow channel enhanced type 12-bit data and 8-bit ID data for SENT signal triggering in decimal notation.	5-312
:HEXa	Sets the slow channel enhanced type 12-bit data and 8-bit ID data for SENT signal triggering in hexadecimal notation.	5-312
:MODE	Sets or queries the data enable/disable condition of the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal triggering.	5-312
:PATtern	Sets the slow channel enhanced type 12-bit data and 8-bit ID data for SENT signal triggering in binary notation.	5-312
:ID?	Queries all ID settings related to the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal triggering.	5-313
:ID		
:CONDition	Sets or queries the ID comparison condition of slow channel enhanced type 12-bit data and 8-bit ID for SENT signal triggering.	5-313
:DECimal<x>	Sets or queries the ID of the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal triggering in decimal notation.	5-313
:MODE	Sets or queries the ID enable/disable condition of the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal triggering.	5-313
:PFOrmat	Sets the data input format, which is one of the slow channel enhanced type 12-bit data and 8-bit ID data conditions, for SENT signal triggering.	5-313
:Dl6Bit?	Queries all slow channel enhanced type 16-bit data and 4-bit ID settings for SENT signal triggering.	5-313
:Dl6Bit		
:DATA?	Queries all slow channel enhanced type 16-bit data and 4-bit ID data settings for SENT signal triggering.	5-313
:DATA		
:CONDition	Sets or queries the data comparison condition of slow channel enhanced type 16-bit data and 4-bit ID for SENT signal triggering.	5-314
:DECimal<x>	Sets or queries the slow channel enhanced type 16-bit data and 4-bit ID data for SENT signal triggering in decimal notation.	5-314
:HEXa	Sets the slow channel enhanced type 16-bit data and 4-bit ID data for SENT signal triggering in hexadecimal notation.	5-314
:MODE	Sets or queries the data enable/disable condition of the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal triggering.	5-314
:PATtern	Sets or queries the slow channel enhanced type 16-bit data and 4-bit ID data for SENT signal triggering in binary notation.	5-314
:ID?	Queries all ID settings related to the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal triggering.	5-314
:ID		
:CONDition	Sets or queries the ID comparison condition of slow channel enhanced type 16-bit data and 4-bit ID for SENT signal triggering.	5-315
:DECimal<x>	Sets or queries the ID of the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal triggering in decimal notation.	5-315
:MODE	Sets or queries the ID enable/disable condition of the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal triggering.	5-315
:PFOrmat	Sets the data input format, which is one of the slow channel enhanced type 16-bit data and 4-bit ID data conditions, for SENT signal triggering.	5-315
:SHORT?	Queries all slow channel short type settings for SENT signal triggering.	5-315
:SHORT		
:DATA?	Queries all slow channel short type data settings for SENT signal triggering.	5-315
:DATA		
:CONDition	Sets or queries the short type data comparison condition of the SENT signal trigger slow channel.	5-315
:DECimal<x>	Sets or queries the SENT signal trigger slow channel short type data in decimal notation.	5-316
:HEXa	Sets the SENT signal trigger slow channel short type data in hexadecimal notation.	5-316
:MODE	Sets or queries the short type data enable/disable condition of the SENT signal trigger slow channel.	5-316
:PATtern	Sets or queries the SENT signal trigger slow channel short type data in binary notation.	5-316
:ID?	Queries all slow channel short type ID settings for SENT signal triggering.	5-316

5.1 List of Commands

Command	Function	Page
:ID		
:CONDition	Sets or queries the short type ID comparison condition of the SENT signal trigger slow channel.	5-316
:DECimal<x>	Sets or queries the SENT signal trigger slow channel short type ID in decimal notation.	5-317
:MODE	Sets or queries the short type ID enable/disable condition of the SENT signal trigger slow channel.	5-317
:PFORmat	Sets or queries the data input format, which is one of the slow channel short type data conditions, for SENT signal triggering.	5-317
:SOURce	Sets or queries the SENT signal trigger source.	5-317
:STYPe	Sets or queries the SENT signal trigger slow channel format.	5-317
:USETup?	Queries all settings related to user-defined data of the SENT signal trigger fast channel.	5-317
:USETup		
:DATA<x>?	Queries all settings related to user-defined data of the SENT signal trigger fast channel.	5-317
:DATA<x>		
:MODE	Sets or queries the enable/disable condition of the specified user-defined data of the SENT signal trigger fast channel.	5-318
:ORDer	Sets or queries the byte order of the specified user-defined data of the SENT signal trigger fast channel.	5-318
:SIZE	Sets or queries the data size of the specified user-defined data of the SENT signal trigger fast channel.	5-318
:MULTiplexing	Sets or queries the multiplexing enable/disable condition of the specified user-defined data of the SENT signal trigger fast channel.	5-318
:VERSion	Sets or queries the SENT signal trigger specification version.	5-318
:SIMPlE?	Queries all edge trigger settings.	5-318
:SIMPlE		
:HFRejection	Sets or queries the on/off state of the low-pass filter (HF rejection) for the edge trigger source.	5-319
:LEVel	Sets or queries the trigger level of the edge trigger source.	5-319
:NREJection	Sets or queries the noise rejection of the edge trigger source level.	5-319
:PROBe	Sets or queries the probe of the external edge trigger source.	5-319
:RANGe	Sets or queries the input range of the external edge trigger source.	5-319
:SLOPe	Sets or queries the trigger slope (polarity if windows is ON) of the edge trigger source.	5-319
:SOURce	Sets or queries the edge trigger source.	5-320
:SPATtern?	Queries all user-defined bus signal trigger settings.	5-320
:SPATtern		
:BITSize	Sets or queries the user-defined bus signal trigger bit length.	5-320
:BRATe	Sets or queries the user-defined bus signal trigger bit rate.	5-320
:CLOCK?	Queries all user-defined bus signal trigger clock signal settings.	5-320
:CLOCK		
:MODE	Sets or queries the enable/disable condition of the user-defined bus signal trigger clock signal.	5-320
:POLarity	Sets or queries the polarity of the user-defined bus signal trigger clock signal.	5-320
:SOURce	Sets or queries the user-defined bus signal trigger clock signal.	5-321
:CS?	Queries all user-defined bus signal trigger chip select signal settings.	5-321
:CS		
:ACTive	Sets or queries the active state of the user-defined bus signal trigger chip select signal.	5-321
:SOURce	Sets or queries the user-defined bus signal trigger chip select signal.	5-321
:DATA?	Queries all user-defined bus signal trigger data signal settings.	5-321
:DATA		
:ACTive	Sets or queries the active state of the user-defined bus signal trigger data signal.	5-321
:SOURce	Sets or queries the user-defined bus signal trigger data signal.	5-322
:HEXa	Queries the data conditions of the user-defined bus signal trigger in hexadecimal notation.	5-322
:LATCh?	Queries all user-defined bus signal trigger latch signal settings.	5-322
:LATCh		
:POLarity	Sets or queries the polarity of the user-defined bus signal trigger latch signal.	5-322
:SOURce	Sets or queries the user-defined bus signal trigger latch signal.	5-322

5.1 List of Commands

Command	Function	Page
:PATtern	Sets or queries the data conditions of the user-defined bus signal trigger in binary notation.	5-322
:SPI?	Queries all SPI bus signal trigger settings.	5-322
:SPI		
:BITorder	Sets or queries the bit order of the SPI bus signal trigger data.	5-322
:CLOCK?	Queries all SPI bus signal trigger clock signal settings.	5-323
:CLOCK		
:POLarity	Sets or queries the polarity of the SPI bus signal trigger clock signal.	5-323
:SOURce	Sets or queries the SPI bus signal trigger clock signal.	5-323
:CS?	Queries all SPI bus signal trigger chip select signal settings.	5-323
:CS		
:ACTive	Sets or queries the active state of the SPI bus signal trigger chip select signal.	5-323
:SOURce	Sets or queries the SPI bus signal trigger chip select signal.	5-323
:DATA<x>?	Queries all SPI bus signal trigger data signal settings.	5-323
:DATA<x>		
:BCount	Sets or queries the pattern comparison start position of the specified data for SPI bus signal triggering.	5-323
:CONDition	Sets or queries the judgment method (true/false) of the specified data for SPI bus signal triggering.	5-324
:DBYTe	Sets or queries the data size (number of bytes) of the specified data for SPI bus signal triggering.	5-324
:DPATtern?	Queries all SPI bus signal trigger data settings.	5-324
:DPATtern		
:HEXa	Sets the specified data for SPI bus signal triggering in hexadecimal notation.	5-324
:PATtern	Sets or queries the specified data for SPI bus signal triggering in binary notation.	5-324
:MODE	Sets or queries the enable/disable condition of data signal for SPI bus signal triggering.	5-324
:SOURce	Sets or queries the SPI bus signal trigger data signal.	5-324
:MODE	Sets or queries the wiring system (three-wire system or four-wire system) of the SPI bus trigger.	5-325
:TYPE	Sets or queries the trigger type.	5-325
:UART?	Queries all UART signal trigger settings.	5-325
:UART		
:BITorder	Sets or queries the UART signal trigger bit order.	5-325
:BRATe	Sets or queries the UART signal trigger bit rate (data transfer rate).	5-325
:DATA?	Queries all UART signal trigger data settings.	5-325
:DATA		
:ASCii	Sets the UART signal trigger data in ASCII format.	5-325
:CONDition	Sets or queries the data judgment method (true/false) for UART signal triggering.	5-325
:CSENSitive	Sets or queries whether to distinguish uppercase and lowercase letters in ASCII data for UART signal triggering.	5-326
:DBYTe	Sets or queries the number of data bytes for UART signal triggering.	5-326
:DPATtern?	Queries all UART signal trigger data settings.	5-326
:DPATtern		
:HEXa	Sets the data for UART signal triggering in hexadecimal notation.	5-326
:PATtern	Sets or queries the data for UART signal triggering in binary notation.	5-326
:PFORMat	Sets or queries the data input format, which is one of the ID and data conditions, for UART signal triggering.	5-326
:ERRor?	Queries all UART signal trigger error settings.	5-326
:ERRor		
:FRAMing	Sets or queries the UART signal trigger framing error setting.	5-326
:PARity	Sets or queries the UART signal trigger parity error setting.	5-327
:PMODE	Sets or queries the UART signal trigger parity mode setting.	5-327
:FORMat	Sets or queries the UART signal trigger format setting.	5-327
:MODE	Sets or queries the trigger type for UART signal triggering.	5-327
:POLarity	Sets or queries the UART signal analysis polarity.	5-327
:SOURce	Sets or queries the UART signal analysis signal.	5-327
:SPOint	Sets or queries the UART signal analysis sample point.	5-327
:WINDow?	Queries all window trigger settings.	5-327
:WINDow		

5.1 List of Commands

Command	Function	Page
:POLarity	Sets or queries the polarity of the window trigger source.	5-328
:SOURce	Sets or queries the window trigger source.	5-328
:TQQualify	Sets or queries the time condition for window triggering.	5-328
:WOR?	Queries all window OR trigger settings.	5-328
:WOR		
:ALL	Sets or queries whether to make all channels window OR trigger sources at once.	5-328
:CHANnel<x>	Sets or queries the slope of each channel of the window OR trigger.	5-328
[:ATRigger]		
:INTERval?	Queries all edge interval trigger settings.	5-328
:INTERval		
:SLOPe	Sets or queries the edge interval trigger slope.	5-328
:SOURce	Sets or queries the edge interval trigger source.	5-328
:TQQualify	Sets or queries the time condition for edge interval triggering.	5-329
:PULSe?	Queries all pulse width trigger settings.	5-329
:PULSe		
:POLarity	Sets or queries the pulse width trigger source polarity.	5-329
:SOURce	Sets or queries the pulse width trigger source.	5-329
:TQQualify	Sets or queries the time condition for pulse width triggering.	5-329
:RFTime?	Queries all rise/fall time trigger settings.	5-329
:RFTime		
:SLOPe	Sets or queries the rise/fall time trigger slope.	5-329
:SOURce	Sets or queries the rise/fall time trigger source.	5-329
:TQQualify	Sets or queries the time condition for rise/fall time triggering.	5-329
:RUNT?	Queries all runt interval trigger settings.	5-330
:RUNT		
:POLarity	Sets or queries the polarity of the runt trigger source.	5-330
:SOURce	Sets or queries the runt trigger source.	5-330
:TQQualify	Sets or queries the time condition for runt triggering.	5-330
:TIMEout?	Queries all timeout trigger settings.	5-330
:TIMEout		
:POLarity	Sets or queries the polarity of the timeout period trigger source.	5-330
:SOURce	Sets or queries the timeout period trigger source.	5-330
:TV?	Queries all TV trigger settings.	5-330
:TV		
:FIELD	Sets or queries the field that the TV trigger is to be activated on.	5-330
:FRAME	Sets or queries the TV trigger frame skip function.	5-330
: {HDTV NTSC PAL SDTV USERdefine}?	Queries all TV trigger mode settings.	5-331
: {HDTV NTSC PAL SDTV USERdefine}		
:LINE	Sets or queries the line that the TV trigger is to be activated on.	5-331
:POLarity	Sets or queries the TV trigger input polarity.	5-331
:LEVel	Sets or queries the TV trigger level.	5-331
:LFORMat	Sets or queries the line designation method for TV triggering.	5-331
:SOURce	Sets or queries the TV trigger source.	5-331
:TYPE	Sets or queries the TV trigger input type.	5-331
:USERdefine?	Queries all user-defined TV trigger settings.	5-331
:USERdefine		
:DEFinition	Sets or queries the user-defined TV trigger definition.	5-331
:HFRejection	Sets or queries the on/off state of the low-pass filter (HF rejection) for user-defined TV triggering.	5-332
:HSYNc	Sets or queries the horizontal sync signal frequency for user-defined TV triggering.	5-332
:SGUard	Sets or queries the sync guard for user-defined TV triggering.	5-332
:COMBination	Sets or queries the trigger combination.	5-332
:DELaY?	Queries all delay settings.	5-332
:DELaY		
:TIME	Sets or queries the delay (time from the trigger point to the trigger position).	5-332
:DETect?	Queries whether a trigger has occurred.	5-333
:FORCe	Forces the instrument to trigger.	5-333
:HOLDoff?	Queries all hold-off settings.	5-333

5.1 List of Commands

Command	Function	Page
:HOLDOff		
:TIME	Sets or queries the hold-off time.	5-333
:MODE	Sets or queries the trigger mode.	5-333
:POSition	Sets or queries the trigger position.	5-333
:SCOUNT	Sets or queries the number of times the trigger condition is to be met when the trigger mode is set to N Single.	5-333
:SOURce?	Queries all enhanced trigger source settings.	5-333
:SOURce		
:CHANnel<x>?	Queries all settings related to the specified trigger source for enhanced triggering.	5-333
:CHANnel<x>		
:HFRejection	Sets or queries the low-pass filter of the specified trigger source for enhanced triggering.	5-334
:LEVel	Sets or queries the trigger level of the specified trigger source for enhanced triggering.	5-334
:NREJection	Sets or queries the noise rejection of the specified trigger source for enhanced triggering.	5-334
:ULLeVel	Sets or queries the upper and lower limits of the enhanced trigger source level.	5-334
:TIME<x>	Sets or queries the time for enhanced triggering.	5-334

WAVEform Group

:WAVEform?	Queries all waveform data output settings.	5-335
:WAVEform		
:ALL?	Queries all settings related to the :WAVEform:ALL:SEND? query.	5-335
:ALL		
:SEND?	Queries the waveform data specified by :WAVEform:ALL:TRACe.	5-335
:TRACe	Sets or queries the target waveform for the :WAVEform:ALL:SEND? command.	5-336
:BITS?	Queries the bit length of the waveform data specified by :WAVEform:TRACe.	5-336
:BYTeorder	Sets or queries the transmission byte order for waveform data in word format, which is 2 bytes or longer in length.	5-336
:END	Sets or queries the end data point in the waveform specified by :WAVEform:TRACe.	5-336
:FORMat	Sets or queries the format of the waveform data to be sent.	5-336
:LENGth?	Queries the total number of points of the waveform specified by :WAVEform:TRACe.	5-336
:OFFSet?	Queries the offset value used to convert the waveform data specified by :WAVEform:TRACe to physical values.	5-336
:POSition?	Queries the vertical position used to convert waveform data into voltage when :WAVEform:FORMat is set to RBYTe.	5-336
:RANGe?	Queries the range value used to convert the waveform data specified by :WAVEform:TRACe to physical values.	5-336
:RECORD	Sets or queries the record number that WAVEform commands will be applied to.	5-337
:RECORD? MINimum	Queries the smallest record number in the source channel history.	5-337
:SEND?	Queries the waveform data specified by :WAVEform:TRACe.	5-337
:SIGN?	Queries whether signs are included in the binary data of the source waveform specified by :WAVEform:TRACe when the data is queried.	5-337
:SRAtE?	Queries the sample rate of the record specified by :WAVEform:RECORD.	5-337
:START	Sets or queries the start data point in the waveform specified by :WAVEform:TRACe.	5-338
:TRACe	Sets or queries the waveform that WAVEform commands will be applied to.	5-338
:TRIGger?	Queries the trigger position of the record specified by :WAVEform:RECORD.	5-338
:TYPE?	Queries the acquisition mode of the waveform specified by :WAVEform:TRACe.	5-338

WPARameter Group

:WPARameter<x>?	Queries all waveform parameter measurement settings.	5-339
:WPARameter<x>		
:DISPlay	Sets or queries the on/off state of waveform parameter measurement display.	5-339
:HISTogram?	Queries all settings related to the histogram display of waveform parameter measurement.	5-339

5.1 List of Commands

Command	Function	Page
:HISTogram		
:MEASure?	Queries all settings related to the automated measurement on the histogram display of waveform parameter measurement.	5-339
:MEASure		
:MODE	Sets or queries the auto measurement mode of the histogram display of waveform parameter measurement.	5-339
:PARAmeter?	Queries all settings related to the automated measurement of the histogram parameters of waveform parameter measurement.	5-339
:PARAmeter		
:ALL	Collectively turns on or off all the histogram parameters of waveform parameter measurement.	5-339
:<Parameter>?	Queries all settings related to the histogram parameters of waveform parameter measurement.	5-339
:<Parameter>		
:STATe	Sets or queries the on/off state the histogram parameters of waveform parameter measurement display.	5-339
:VALue?	Queries the measured value of a histogram parameter of waveform parameter measurement.	5-340
:POSition<y>	Sets or queries the position a parameter of waveform parameter measurement histogram.	5-340
:ITEM	Sets or queries a waveform parameter of waveform parameter measurement.	5-340
:MODE	Sets or queries the waveform parameter measurement mode.	5-340
:TREND?	Queries all settings related to the trend display of waveform parameter measurement.	5-340
:TREND		
:AScale	Executes auto scaling of the trend display of waveform parameter measurement.	5-341
:CURSor?	Queries all settings related to the cursor measurement on the trend display of waveform parameter measurement.	5-341
:CURSor		
:C<y>?	Queries the measured value of a cursor of waveform parameter measurement.	5-341
:DC?	Queries the measured value of a cursor of the trend of waveform parameter measurement.	5-341
:MODE	Sets or queries the auto measurement mode of the trend of waveform parameter measurement.	5-341
:POSition<y>	Sets or queries the position a cursor of the trend of waveform parameter measurement display.	5-341
:HRANge	Sets or queries the trend display source window for waveform parameter measurement.	5-341
:HSPan	Sets or queries the trend display horizontal span for waveform parameter measurement.	5-341
:VERTical	Sets or queries the vertical range of the trend of waveform parameter measurement.	5-341
:VTDisplay	Sets or queries the VT waveform display on/off state.	5-342
:WAIT?	Waits for the completion of automated measurement with a timeout.	5-342

XY Group

:XY<x>?	Queries all XY display settings.	5-343
:XY<x>		
:DISPlay	Sets or queries whether to show (ON) or hide (OFF) the XY display on the screen.	5-343
:MEASure?	Queries all automated measurement settings of XY display.	5-343
:MEASure		
:CURSor?	Queries all cursor measurement settings of XY display.	5-343
:CURSor		
:DX?	Queries all settings for the voltage difference between the XY display's horizontal cursors.	5-343
:DX		
:STATe	Sets or queries the on/off status of the voltage difference between the XY display's horizontal cursors.	5-343
:VALue?	Queries the voltage difference between the XY display's horizontal cursors.	5-343

5.1 List of Commands

Command	Function	Page
:DY?	Queries all settings for the voltage difference between the XY display's vertical cursors.	5-343
:DY		
:STATe	Sets or queries the on/off status of the voltage difference between the XY display's vertical cursors.	5-343
:VALue?	Queries the voltage difference between the XY display's vertical cursors.	5-343
:X<y>?	Queries all horizontal cursor settings of XY display.	5-343
:X<y>		
:POSition	Sets or queries a horizontal cursor position on the XY display.	5-344
:STATe	Sets or queries the on/off status of the voltage of an XY display's horizontal cursor.	5-344
:VALue?	Queries the voltage at a XY display's horizontal cursor.	5-344
:Y<y>?	Queries all vertical cursor settings of XY display.	5-344
:Y<y>		
:POSition	Sets or queries a vertical cursor position on the XY display.	5-344
:STATe	Sets or queries the on/off status of the voltage of an XY display's vertical cursor.	5-344
:VALue?	Queries the voltage at a XY display's vertical cursor.	5-344
:INTeg?	Queries all integral settings of XY display.	5-344
:INTeg		
:LOOP	Sets or queries the integral method on the XY display.	5-344
:POLarity	Sets or queries the integral direction on the XY display.	5-345
:VALue?	Queries the integral value on the XY display.	5-345
:MODE	Sets or queries the automated measurement mode on the XY display.	5-345
:RANGe	Sets or queries the VT waveform source window for the XY display.	5-345
:SPLit	Sets or queries whether the XY1/XY2 display screen is split on the XY display.	5-345
:TRANGe	Sets or queries the VT waveform range for the XY display.	5-345
:VTDisplay	Sets or queries the on/off state of the VT waveform display on the XY display.	5-345
:XTRace	Sets or queries the channel that is assigned to the X-axis on the XY display.	5-345
:YTRace	Sets or queries the channel that is assigned to the Y-axis on the XY display.	5-345

ZOOM Group

:ZOOM<x>?	Queries all waveform zoom settings.	5-346
:ZOOM<x>		
:ALLocation?	Queries all zoom source waveform settings.	5-346
:ALLocation		
: {CHANnel<y> MATH<y>}	Sets or queries the zoom source waveform.	5-346
:DISPlay	Sets or queries the zoom waveform display on/off state.	5-346
:FORMat	Sets or queries the zoom waveform display format.	5-346
:MAG	Sets or queries the magnification of a zoom waveform display window.	5-346
:MAGFine	Sets or queries the magnification (fine) of a zoom waveform display window.	5-346
:MAIN	Sets or queries the zoom waveform display ratio.	5-346
:POSition	Sets or queries the position of a zoom box.	5-346
:VERTical?	Queries all vertical zoom settings.	5-346
:VERTical		
: {CHANnel<y> MATH<y>}?	Queries all settings related to a vertical zoom trace.	5-347
: {CHANnel<y> MATH<y>}		
:MAG	Sets or queries the vertical zoom factor.	5-347
:POSition	Sets or queries the vertical zoom position.	5-347
:RESet	Resets all vertical zoom settings.	5-347
:TRACe	Sets or queries the trace to show on the vertical zoom window.	5-347

Common Command Group

*CAL?	Executes calibration and queries the result.	5-348
*CLS	Clears the standard event register, extended event register, and error queue.	5-348
*ESE	Sets or queries the standard event enable register.	5-348
*ESR?	Queries and clears the standard event register.	5-348
*IDN?	Queries the instrument model.	5-348
*OPC	Sets bit 0 (the OPC bit) of the standard event register to 1 upon the completion of the specified overlap command.	5-348

5.1 List of Commands

Command	Function	Page
*OPC?	When you send *OPC?, the instrument returns ASCII code 1 when the specified overlap command is completed.	5-349
*OPT?	Queries the installed options.	5-349
*RST	Initializes the settings.	5-349
*SRE	Sets or queries the service request enable register value.	5-349
*STB?	Queries the status byte register value.	5-349
*TST?	Executes a self-test and queries the result. The self-test consists of tests of each kind of internal memory.	5-350
*WAI	Holds the execution of the subsequent command until the specified overlap command is completed.	5-350

5.2 ACQuire Group

:ACQuire?

Function Queries all waveform acquisition settings.
Syntax :ACQuire?

:ACQuire:AVERage?

Function Queries all settings related to averaging and waveform acquisition counts.
Syntax :ACQuire:AVERage?

:ACQuire:AVERage:COUNT

Function Sets or queries the attenuation constant of exponential averaging or the average count of simple averaging when in averaging mode.
Syntax :ACQuire:AVERage:COUNT {<NRf>}
<NRf> = 2 to 1024, 2n steps
Example :ACQUIRE:AVERAGE:COUNT 2
:ACQUIRE:AVERAGE:COUNT?
-> :ACQUIRE:AVERAGE:COUNT 2

:ACQuire:COUNT

Function Sets or queries the number of waveform acquisitions to perform in normal, envelope, and averaging modes.
Syntax :ACQuire:COUNT {<NRf>|INFinity}
<NRf> = 1 to 65536
Example :ACQUIRE:COUNT 1
:ACQUIRE:COUNT?
-> :ACQUIRE:COUNT 1

:ACQuire:MODE

Function Sets or queries the waveform acquisition mode.
Syntax :ACQuire:MODE {AVERage|ENVELOpe|NORMAL}
Example :ACQUIRE:MODE AVERAGE
:ACQUIRE:MODE?
-> :ACQUIRE:MODE AVERAGE

:ACQuire:RESolution

Function Sets or queries the high resolution mode on/off state.
Syntax :ACQuire:RESolution {<Boolean>}
Example :ACQUIRE:RESOLUTION ON
:ACQUIRE:RESOLUTION?
-> :ACQUIRE:RESOLUTION 1
Description When the high resolution mode is turned on, the maximum record length is halved.

:ACQuire:RLENgth

Function Sets or queries the record length.
Syntax :ACQuire:RLENgth {<NRf>}
<NRf> = See the Features Guide for this information.
Example :ACQUIRE:RLENGTH 1250
:ACQUIRE:RLENGTH?
-> :ACQUIRE:RLENGTH 1250

:ACQuire:SAMPling

Function Sets or queries the sampling mode.
Syntax :ACQuire:SAMPling {REAL|INTERporate|REPetitive}
Example :ACQUIRE:SAMPLING REAL
:ACQUIRE:SAMPLING?
-> :ACQUIRE:SAMPLING REAL

5.3 ANALysis Group

The power supply analysis feature (:ANALysis:PANalyze and :ANALysis:PMEasure commands) is an option. Power supply analysis commands cannot be used on models without the power supply analysis option.

:ANALysis?

Function Queries all analysis feature settings.
Syntax :ANALysis?

:ANALysis:AHISTogram<x>?

Function Queries all settings related to the histogram of the analysis feature.
Syntax :ANALysis:AHISTogram<x>?
<x> = 1 to 2

:ANALysis:AHISTogram<x>:DISPlay

Function Sets or queries the waveform histogram display on/off state.
Syntax :ANALysis:AHISTogram<x>:
DISPlay {<Boolean>}
:ANALysis:AHISTogram<x>:DISPlay?
<x> = 1 to 2
Example :ANALYSIS:AHISTOGRAM1:DISPLAY ON
:ANALYSIS:AHISTOGRAM1:DISPLAY?
-> :ANALYSIS:AHISTOGRAM1:DISPLAY 1

:ANALysis:AHISTogram<x>:HORizontal

Function Sets or queries the horizontal range of the waveform histogram.
Syntax :ANALysis:AHISTogram<x>:
HORizontal {<NRf>,<NRf>}
:ANALysis:AHISTogram<x>:HORizontal?
<x> = 1 to 2
<NRf> = -4 to 4 (div)
Example :ANALYSIS:AHISTOGRAM1:HORIZONTAL 0,1
:ANALYSIS:AHISTOGRAM1:HORIZONTAL?
-> :ANALYSIS:AHISTOGRAM1:
HORIZONTAL 1.000E+00,0.000E+00

:ANALysis:AHISTogram<x>:MEASure?

Function Queries all settings related to the waveform histogram measurement (including the mode on/off state).
Syntax :ANALysis:AHISTogram<x>:MEASure?
<x> = 1 to 2

:ANALysis:AHISTogram<x>:MEASure:MODE

Function Sets or queries the waveform histogram measurement mode.
Syntax :ANALysis:AHISTogram<x>:MEASure:
MODE {OFF|PARAMeter}
:ANALysis:AHISTogram<x>:MEASure:
MODE?
<x> = 1 to 2
Example :ANALYSIS:AHISTOGRAM1:MEASURE:
MODE OFF
:ANALYSIS:AHISTOGRAM1:MEASURE:MODE?
-> :ANALYSIS:AHISTOGRAM1:MEASURE:
MODE OFF

:ANALysis:AHISTogram<x>:MEASure:PARAMeter?

Function Queries all settings related to the waveform histogram measurement when the mode is set to Param.
Syntax :ANALysis:AHISTogram<x>:MEASure:
PARAMeter?
<x> = 1 to 2

:ANALysis:AHISTogram<x>:MEASure:PARAMeter:ALL

Function Collectively turns on or off all the measurement items of the waveform histogram.
Syntax :ANALysis:AHISTogram<x>:MEASure:
PARAMeter:ALL {<Boolean>}
<x> = 1 to 2
Example :ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:ALL ON

:ANALysis:AHISTogram<x>:MEASure:PARAMeter:<Parameter>?

Function Queries all settings related to the specified measurement item of the waveform histogram.
Syntax :ANALysis:AHISTogram<x>:MEASure:
PARAMeter:<Parameter>?
<x> = 1 to 2
<Parameter> = {C1|C2|DC|MAXimum|MEAN|MEDian|MINimum|PEAK|SD2integ|SD3integ|SDEVIation|SDINtegr}

:ANALysis:AHIStogram<x>:MEASure:PARAMeter:<Parameter>:STATe

Function Sets or queries the on/off state of the specified measurement item of the waveform histogram.

Syntax :ANALysis:AHIStogram<x>:MEASure:
PARAMeter:<Parameter>:
STATe {<Boolean>}
:ANALysis:AHIStogram<x>:MEASure:
PARAMeter:<Parameter>:STATe?
<x> = 1 to 2
<Parameter> = {C1|C2|DC|MAXimum|MEAN|ME
Dian|MINimum|PEAK|SD2integ|SD3integ|SDEVi
ation|SDINteg}

Example The following example is for the maximum value.
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:MAXIMUM:STATE ON
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:MAXIMUM:STATE?
-> :ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:MAXIMUM:STATE 1

:ANALysis:AHIStogram<x>:MEASure:PARAMeter:<Parameter>:VALue?

Function Queries the automatically measured value of the specified measurement item of the waveform histogram.

Syntax :ANALysis:AHIStogram<x>:MEASure:
PARAMeter:<Parameter>:VALue?
<x> = 1 to 2
<Parameter> = {C1|C2|DC|MAXimum|MEAN|ME
Dian|MINimum|PEAK|SD2integ|SD3integ|SDEVi
ation|SDINteg}

Example The following example is for the maximum value.
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:MAXIMUM:VALUE?
-> :ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:MAXIMUM:VALUE 1.000E+00

:ANALysis:AHIStogram<x>:MEASure:PARAMeter:HPOSition<y>

Function Sets or queries the horizontal position of Cursor1 or Cursor2 of the waveform histogram.

Syntax :ANALysis:AHIStogram<x>:MEASure:
PARAMeter:HPOSition<y> {<Nrf>}
:ANALysis:AHIStogram<x>:MEASure:
PARAMeter:HPOSition<y>?
<x> = 1 to 2
<y> = 1 to 2
<Nrf> = -5 to 5 (div)

Example :ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:HPOSITION1 1
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:HPOSITION1?
-> :ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:HPOSITION 1.000E+00

:ANALysis:AHIStogram<x>:MEASure:PARAMeter:VPOSition<y>

Function Sets or queries the vertical position of Cursor1 or Cursor2 of the waveform histogram.

Syntax :ANALysis:AHIStogram<x>:MEASure:
PARAMeter:VPOSition<y> {<Nrf>}
:ANALysis:AHIStogram<x>:MEASure:
PARAMeter:VPOSition<y>?
<x> = 1 to 2
<y> = 1 to 2
<Nrf> = -4 to 4 (div)

Example :ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:VPOSITION1 1
:ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:VPOSITION1?
-> :ANALYSIS:AHISTOGRAM1:MEASURE:
PARAMETER:VPOSITION 1.000E+00

:ANALysis:AHIStogram<x>:MODE

Function Sets or queries the target axis of the waveform histogram.

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

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

:ANALysis:AHIStogram<x>:RANGE

Function Sets or queries the measurement source window of the waveform histogram.

Syntax :ANALysis:AHIStogram<x>:RANGE {MAIN|
Z1|Z2}
:ANALysis:AHIStogram<x>:RANGE?
<x> = 1 to 2

Example :ANALYSIS:AHISTOGRAM1:RANGE MAIN
:ANALYSIS:AHISTOGRAM1:RANGE?
-> :ANALYSIS:AHISTOGRAM1:RANGE MAIN

:ANALysis:AHIStogram<x>:TRACe

Function Sets or queries the source waveform of the waveform histogram.

Syntax :ANALysis:AHIStogram<x>:
TRACe {<Nrf>|MATH<y>}
:ANALysis:AHIStogram<x>:TRACe?
<x> = 1 to 2
<Nrf> = 1 to 4
<y> of MATH<y> = 1 to 4

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

5.3 ANALysis Group

:ANALysis:AHIStoqram<x>:VERTical

Function Sets or queries the vertical range of the waveform histogram.

Syntax :ANALysis:AHIStoqram<x>:
VERTical {<NRf>,<NRf>}
:ANALysis:AHIStoqram<x>:VERTical?
<x> = 1 to 2
<NRf> = -4 to 4 (div)

Example :ANALYSIS:AHISTOGRAM1:VERTICAL 0,1
:ANALYSIS:AHISTOGRAM1:VERTICAL?
-> :ANALYSIS:AHISTOGRAM1:
VERTICAL 1.000E+00,0.000E+00

:ANALysis:PANalyze?

Function Queries all power supply analysis settings.

Syntax :ANALysis:PANalyze?

:ANALysis:PANalyze:HARMonics?

Function Queries all harmonic analysis settings.

Syntax :ANALysis:PANalyze:HARMonics?

:ANALysis:PANalyze:HARMonics:CCLass?

Function Queries all settings related to harmonic analysis class C.

Syntax :ANALysis:PANalyze:HARMonics:CCLass?

:ANALysis:PANalyze:HARMonics:CCLass:GETLambda

Function Sets the current power factor of harmonic analysis class C.

Syntax :ANALysis:PANalyze:HARMonics:CCLass:
GETLambda

Example :ANALYSIS:PANALYZE:HARMONICS:CCLASS:
GETLAMBDA

:ANALysis:PANalyze:HARMonics:CCLass:LAMBda

Function Sets or queries the power factor of harmonic analysis class C.

Syntax :ANALysis:PANalyze:HARMonics:CCLass:
LAMBda {<NRf>}
:ANALysis:PANalyze:HARMonics:CCLass:
LAMBda?

Example :ANALYSIS:PANALYZE:HARMONICS:CCLASS:
LAMBDA 0.10
:ANALYSIS:PANALYZE:HARMONICS:CCLASS:
LAMBDA?
-> :ANALYSIS:PANALYZE:HARMONICS:
CCLASS:LAMBDA 100.0E-03

:ANALysis:PANalyze:HARMonics:CCLass:MAXCurrent

Function Sets or queries the fundamental current value of harmonic analysis class C.

Syntax :ANALysis:PANalyze:HARMonics:CCLass:
MAXCurrent {<NRf>|<Current>}
:ANALysis:PANalyze:HARMonics:CCLass:
MAXCurrent?
<NRf>, <Current> = 0.001 to 100 (A)

Example :ANALYSIS:PANALYZE:HARMONICS:CCLASS:
MAXCURRENT 50A
:ANALYSIS:PANALYZE:HARMONICS:CCLASS:
MAXCURRENT?
-> :ANALYSIS:PANALYZE:HARMONICS:
CCLASS:MAXCURRENT 50.000E+00

:ANALysis:PANalyze:HARMonics:CCLass:OPower

Function Sets or queries whether active power 25 W of harmonic analysis class C will be exceeded.

Syntax :ANALysis:PANalyze:HARMonics:CCLass:
OPower {FALSE|TRUE}
:ANALysis:PANalyze:HARMonics:CCLass:
OPower?

Example :ANALYSIS:PANALYZE:HARMONICS:CCLASS:
OPower FALSE
:ANALYSIS:PANALYZE:HARMONICS:CCLASS:
OPower?
-> :ANALYSIS:PANALYZE:HARMONICS:
CCLASS:OPower FALSE

:ANALysis:PANalyze:HARMonics:CLAss

Function Sets or queries the applicable class of the DUT in harmonic analysis.

Syntax :ANALysis:PANalyze:HARMonics:
CLAss {A|B|C|D}
:ANALysis:PANalyze:HARMonics:CLAss?

Example :ANALYSIS:PANALYZE:HARMONICS:CLASS A
:ANALYSIS:PANALYZE:HARMONICS:CLASS?
-> :ANALYSIS:PANALYZE:HARMONICS:
CLASS A

:ANALysis:PANalyze:HARMonics:DCLass?

Function Queries all settings related to harmonic analysis class D.

Syntax :ANALysis:PANalyze:HARMonics:DCLass?

:ANALysis:PANalyze:HARMonics:DCLass:POWer

Function Sets or queries the power value of harmonic analysis class D.

Syntax :ANALysis:PANalyze:HARMonics:DCLass:POWER {<NRf>}
:ANALysis:PANalyze:HARMonics:DCLass:POWER?
<NRf> = -1.0000E+31 to 1.0000E+31

Example :ANALYSIS:PANALYZE:HARMONICS:DCLASS:POWER 1V
:ANALYSIS:PANALYZE:HARMONICS:DCLASS:POWER?
-> :ANALYSIS:PANALYZE:HARMONICS:DCLASS:POWER 1.0000E+00

:ANALysis:PANalyze:HARMonics:DMODE

Function Sets or queries the display mode of harmonic analysis.

Syntax :ANALysis:PANalyze:HARMonics:DMODE {LINEar|LOG}
:ANALysis:PANalyze:HARMonics:DMODE?

Example :ANALYSIS:PANALYZE:HARMONICS:DMODE LINEAR
:ANALYSIS:PANALYZE:HARMONICS:DMODE?
-> :ANALYSIS:PANALYZE:HARMONICS:DMODE LINEAR

:ANALysis:PANalyze:HARMonics:DETail?

Function Queries all settings related to the harmonic analysis result list.

Syntax :ANALysis:PANalyze:HARMonics:DETail?

:ANALysis:PANalyze:HARMonics:DETail:DISPlay

Function Sets or queries the display mode of the harmonic analysis result list.

Syntax :ANALysis:PANalyze:HARMonics:DETail:DISPlay {FULL|LOWer|UPPer}
:ANALysis:PANalyze:HARMonics:DETail:DISPlay?

Example :ANALYSIS:PANALYZE:HARMONICS:DETAIL:DISPLAY FULL
:ANALYSIS:PANALYZE:HARMONICS:DETAIL:DISPLAY?
-> :ANALYSIS:PANALYZE:HARMONICS:DETAIL:DISPLAY FULL

:ANALysis:PANalyze:HARMonics:DETail:LIST:ITEM?

Function Queries all items that will be displayed in the harmonic analysis result list.

Syntax :ANALysis:PANalyze:HARMonics:DETail:LIST:ITEM?

Example :ANALYSIS:PANALYZE:HARMONICS:DETAIL:LIST:ITEM?
-> :ANALYSIS:PANALYZE:HARMONICS:DETAIL:LIST:ITEM "Order.,Measure(A),Limit(A),Measure(%),Limit(%),Info,"

:ANALysis:PANalyze:HARMonics:DETail:LIST:VALue?

Function Queries all the data for the specified analysis number in the harmonic analysis result list.

Syntax :ANALysis:PANalyze:HARMonics:DETail:LIST:VALue? {<NRf>}
<NRf> = 1 to 40 (harmonic order)

Example :ANALYSIS:PANALYZE:HARMONICS:DETAIL:LIST:VALUE? 2
-> :ANALYSIS:PANALYZE:HARMONICS:DETAIL:LIST:VALUE " 2, 0.031, 0.020, 3.149, 2.000,NG,"

:ANALysis:PANalyze:HARMonics:GROUping

Function Sets or queries the harmonic analysis grouping.

Syntax :ANALysis:PANalyze:HARMonics:GROUping {OFF|TYPE1|TYPE2}
:ANALysis:PANalyze:HARMonics:GROUping?

Example :ANALYSIS:PANALYZE:HARMONICS:GROUPING OFF
:ANALYSIS:PANALYZE:HARMONICS:GROUPING?
-> :ANALYSIS:PANALYZE:HARMONICS:GROUPING OFF

:ANALysis:PANalyze:HARMonics:IECEditionon

Function Sets or queries measurement standard (IEC 61000-4-7) of the harmonic analysis measurement mode.

Syntax :ANALysis:PANalyze:HARMonics:IECEdition {ED2_0|ED2_1}
:ANALysis:PANalyze:HARMonics:IECEdition?

Example :ANALYSIS:PANALYZE:HARMONICS:IECEDITION ED2_0
:ANALYSIS:PANALYZE:HARMONICS:IECEDITION?
-> :ANALYSIS:PANALYZE:HARMONICS:IECEDITION ED2_0

5.3 ANALysis Group

:ANALysis:PANalyze:HARMonics:RMS

Function Queries the rms value harmonic analysis.
Syntax :ANALysis:PANalyze:HARMonics:RMS?
Example :ANALYSIS:PANALYZE:HARMONICS:RMS?

:ANALysis:PANalyze:HARMonics:SPOint

Function Sets or queries the computation start point of harmonic analysis.
Syntax :ANALysis:PANalyze:HARMonics:SPOint {<NRf>}
:ANALysis:PANalyze:HARMonics:SPOint?<NRf> = -5 to 5 (div)
Example :ANALYSIS:PANALYZE:HARMONICS:SPOINT 1
:ANALYSIS:PANALYZE:HARMONICS:SPOINT?
-> :ANALYSIS:PANALYZE:HARMONICS:SPOINT 1.000E+00

:ANALysis:PANalyze:HARMonics:SVOLTage

Function Sets or queries the supply voltage of harmonic analysis.
Syntax :ANALysis:PANalyze:HARMonics:SVOLTage {<NRf>|<Voltage>}
:ANALysis:PANalyze:HARMonics:SVOLTage?
<NRf>, <Voltage> = 90 V to 440 V
Example :ANALYSIS:PANALYZE:HARMONICS:SVOLTAGE 230
:ANALYSIS:PANALYZE:HARMONICS:SVOLTAGE?
-> :ANALYSIS:PANALYZE:HARMONICS:SVOLTAGE 230.00000E+00

:ANALysis:PANalyze:HARMonics:THD

Function Queries the harmonic distortion factor of harmonic computation.
Syntax :ANALysis:PANalyze:HARMonics:THD?
Example :ANALYSIS:PANALYZE:HARMONICS:THD?

:ANALysis:PANalyze:I2T?

Function Queries all Joule integral settings.
Syntax :ANALysis:PANalyze:I2T?

:ANALysis:PANalyze:I2T:MATH

Function Sets or queries the Joule integral waveform display on/off state.
Syntax :ANALysis:PANalyze:I2T:MATH {I2T|OFF}
:ANALysis:PANalyze:I2T:MATH?
Example :ANALYSIS:PANALYZE:I2T:MATH I2T
:ANALYSIS:PANALYZE:I2T:MATH?
-> :ANALYSIS:PANALYZE:I2T:MATH I2T

:ANALysis:PANalyze:I2T:MEASure?

Function Queries all settings related to the automated measurement of the Joule integral.
Syntax :ANALysis:PANalyze:I2T:MEASure?

:ANALysis:PANalyze:I2T:MEASure:I2T?

Function Queries Joule integral settings.
Syntax :ANALysis:PANalyze:I2T:MEASure:I2T?

:ANALysis:PANalyze:I2T:MEASure:I2T:COU Nt?

Function Queries the continuous statistical processing count of the Joule integral.
Syntax :ANALysis:PANalyze:I2T:MEASure:I2T:COUNT?
Example :ANALYSIS:PANALYZE:I2T:MEASURE:I2T:COUNT?
-> :ANALYSIS:PANALYZE:I2T:MEASURE:I2T:COUNT 100

:ANALysis:PANalyze:I2T:MEASure:I2T:{MA Ximum|MEAN|MINimum|SDEVIation)?

Function Queries the Joule integral statistics.
Syntax :ANALysis:PANalyze:I2T:MEASure:I2T:{MAXimum|MEAN|MINimum|SDEVIation)?
Example :ANALYSIS:PANALYZE:I2T:MEASURE:I2T:MAXIMUM?
-> :ANALYSIS:PANALYZE:I2T:MEASURE:I2T:MAXIMUM 10.0000E+03

Description If the statistical value cannot be obtained, the instrument returns "NAN" (not a number).

:ANALysis:PANalyze:I2T:MEASure:I2T:STA Te

Function Sets or queries whether the Joule integral is to be measured.
Syntax :ANALysis:PANalyze:I2T:MEASure:I2T:STATE {<Boolean>}
:ANALysis:PANalyze:I2T:MEASure:I2T:STATE?
Example :ANALYSIS:PANALYZE:I2T:MEASURE:I2T:STATE ON
:ANALYSIS:PANALYZE:I2T:MEASURE:I2T:STATE?
-> :ANALYSIS:PANALYZE:I2T:MEASURE:I2T:STATE 1

:ANALysis:PANalyze:I2T:MEASure:I2T:VALue?

Function	Queries automatically measured value of the Joule integral.
Syntax	:ANALysis:PANalyze:I2T:MEASure:I2T:VALue? [{<NRf>}] <NRf> = See the Features Guide for this information.
Example	:ANALYSIS:PANALYZE:I2T:MEASURE:I2T:VALUE? -> :ANALYSIS:PANALYZE:I2T:MEASURE:I2T:VALUE 10.0000E+03
Description	- If the value is immeasurable, the instrument returns "NAN" (not a number). - The <NRf> is used to specify which iteration of automated measurement to query the measured value from. - If <NRf> is set to 1, the oldest measured value in the automated measurement memory is queried. - If a measured value does not exist at the specified iteration, the instrument returns "NAN" (not a number). - If <NRf> is omitted, the most recent measured value is queried. - If cyclic statistical processing for automated measurement is being executed and <NRf> is specified, the measured values over a cycle in iteration <NRf> from the left of the display on the displayed waveform is queried. If <NRf> is omitted, the measured values over the last cycle on the displayed waveform is queried.

:ANALysis:PANalyze:I2T:RANGe

Function	Sets or queries the measurement source window.
Syntax	:ANALysis:PANalyze:I2T:RANGe {MAIN Z1 Z2} :ANALysis:PANalyze:I2T:RANGe?
Example	:ANALYSIS:PANALYZE:I2T:RANGE MAIN :ANALYSIS:PANALYZE:I2T:RANGE? -> :ANALYSIS:PANALYZE:I2T:RANGE MAIN

:ANALysis:PANalyze:I2T:SCALE?

Function	Queries all scaling settings.
Syntax	:ANALysis:PANalyze:I2T:SCALE?

:ANALysis:PANalyze:I2T:SCALE:CENTer

Function	Sets or queries the center value for manual scaling.
Syntax	:ANALysis:PANalyze:I2T:SCALE:CENTer {<NRf>} :ANALysis:PANalyze:I2T:SCALE:CENTer?<NRf> = -1.0000E+31 to 1.0000E+31
Example	:ANALYSIS:PANALYZE:I2T:SCALE:CENTER 1 :ANALYSIS:PANALYZE:I2T:SCALE:CENTER? -> :ANALYSIS:PANALYZE:I2T:SCALE:CENTER 1.00000E+00

:ANALysis:PANalyze:I2T:SCALE:MODE

Function	Sets or queries the scale mode.
Syntax	:ANALysis:PANalyze:I2T:SCALE:MODE {AUTO MANual} :ANALysis:PANalyze:I2T:SCALE:MODE?
Example	:ANALYSIS:PANALYZE:I2T:SCALE:MODE AUTO :ANALYSIS:PANALYZE:I2T:SCALE:MODE? -> :ANALYSIS:PANALYZE:I2T:SCALE:MODE AUTO

:ANALysis:PANalyze:I2T:SCALE:SENSitivity

Function	Sets or queries the sensitivity for manual scaling.
Syntax	:ANALysis:PANalyze:I2T:SCALE:SENSitivity {<NRf>} :ANALysis:PANalyze:I2T:SCALE:SENSitivity?<NRf> = 1.0000E-31 to 1.0000E+31
Example	:ANALYSIS:PANALYZE:I2T:SCALE:SENSITIVITY 10 :ANALYSIS:PANALYZE:I2T:SCALE:SENSITIVITY? -> :ANALYSIS:PANALYZE:I2T:SCALE:SENSITIVITY 10.0000E+00

:ANALysis:PANalyze:I2T:TRANGe (TimeRange)

Function	Sets or queries the measurement time period.
Syntax	:ANALysis:PANalyze:I2T:TRANGe {<NRf>,<NRf>} :ANALysis:PANalyze:I2T:TRANGe?<NRf> = -5 to 5 (div), (10 div/display record length) steps
Example	:ANALYSIS:PANALYZE:I2T:TRANGE -4,4 :ANALYSIS:PANALYZE:I2T:TRANGE? -> :ANALYSIS:PANALYZE:I2T:TRANGE -4.00E+00,4.00E+00

5.3 ANALysis Group

:ANALysis:PANalyze:SETup?

Function Queries all power supply analysis input settings.
Syntax :ANALysis:PANalyze:SETup?

:ANALysis:PANalyze:SETup:ADESkew

Function Executes auto deskewing for power supply analysis.
Syntax :ANALysis:PANalyze:SETup:ADESkew
Example :ANALYSIS:PANALYZE:SETUP:ADESKEW

:ANALysis:PANalyze:SETup:I?

Function Queries all current input channel settings for power supply analysis.
Syntax :ANALysis:PANalyze:SETup:I?

:ANALysis:PANalyze:SETup:I:DESKew

Function Sets or queries auto deskewing of the current input channel for power supply analysis.
Syntax :ANALysis:PANalyze:SETup:I:DESKew {<Time>}
:ANALysis:PANalyze:SETup:I:DESKew? <Time> = -1000.00 ns to 1000.00 ns (10 ps steps)
Example :ANALYSIS:PANALYZE:SETUP:I:DESKew 1NS
:ANALYSIS:PANALYZE:SETUP:I:DESKew? -> :ANALYSIS:PANALYZE:SETUP:I:DESKew 1.000E-09

:ANALysis:PANalyze:SETup:I:INPut

Function Sets or queries the current input channels for power supply analysis.
Syntax :ANALysis:PANalyze:SETup:I:INPut {2|4}
:ANALysis:PANalyze:SETup:I:INPut?
Example :ANALYSIS:PANALYZE:SETUP:I:INPUT 2
:ANALYSIS:PANALYZE:SETUP:I:INPUT? -> :ANALYSIS:PANALYZE:SETUP:I:INPUT 2

:ANALysis:PANalyze:SETup:I:PROBe

Function Sets or queries the current-to-voltage conversion ratio settings of the probe connected to the current input channel for power supply analysis.
Syntax :ANALysis:PANalyze:SETup:I:PROBe {C0 _ 001|C0 _ 002|C0 _ 005|C0 _ 01|C0 _ 02|C0 _ 05|C0 _ 1|C0 _ 2|C0 _ 5|C1|C2|C5|C10|C20|C50|C100|C200|C500|C1000|C2000}
:ANALysis:PANalyze:SETup:I:PROBe?
Example :ANALYSIS:PANALYZE:SETUP:I:PROBE C0 _ 001
:ANALYSIS:PANALYZE:SETUP:I:PROBE? -> :ANALYSIS:PANALYZE:SETUP:I:PROBE C0 _ 001

:ANALysis:PANalyze:SETup:U?

Function Queries all voltage input channel settings for power supply analysis.
Syntax :ANALysis:PANalyze:SETup:U?

:ANALysis:PANalyze:SETup:U:DESKew

Function Sets or queries auto deskewing of the voltage input channel for power supply analysis.
Syntax :ANALysis:PANalyze:SETup:U:DESKew {<Time>}
:ANALysis:PANalyze:SETup:U:DESKew? <Time> = -1000.00 ns to 1000.00 ns (10 ps steps)
Example :ANALYSIS:PANALYZE:SETUP:U:DESKew 1NS
:ANALYSIS:PANALYZE:SETUP:U:DESKew? -> :ANALYSIS:PANALYZE:SETUP:U:DESKew 1.000E-09

:ANALysis:PANalyze:SETup:U:INPut

Function Sets or queries the voltage input channels for power supply analysis.
Syntax :ANALysis:PANalyze:SETup:U:INPut {1|3}
:ANALysis:PANalyze:SETup:U:INPut?
Example :ANALYSIS:PANALYZE:SETUP:U:INPUT 1
:ANALYSIS:PANALYZE:SETUP:U:INPUT? -> :ANALYSIS:PANALYZE:SETUP:U:INPUT 1

:ANALysis:PANalyze:SETup:U:PROBe

Function Sets or queries the probe attenuation of the voltage input channels for power supply analysis.

Syntax :ANALysis:PANalyze:SETup:U:
PROBe {<NRf>}
:ANALysis:PANalyze:SETup:U:PROBe?
<NRf> = See the Features Guide for this information.

Example :ANALYSIS:PANALYZE:SETUP:U:PROBE 1
:ANALYSIS:PANALYZE:SETUP:U:PROBE?
-> :ANALYSIS:PANALYZE:SETUP:U:
PROBE 1.000

:ANALysis:PANalyze:SETup:RTRace

Function Sets or queries the deskewing source trace for power supply analysis.

Syntax :ANALysis:PANalyze:SETup:RTRace {I|
U}
:ANALysis:PANalyze:SETup:RTRace?

Example :ANALYSIS:PANALYZE:SETUP:RTRACE I
:ANALYSIS:PANALYZE:SETUP:RTRACE?
-> :ANALYSIS:PANALYZE:SETUP:RTRACE I

:ANALysis:PANalyze:SOA?

Function Queries all settings related to the XY display for safe operating area analysis.

Syntax :ANALysis:PANalyze:SOA?

:ANALysis:PANalyze:SOA:CURSOr?

Function Queries all settings related to the cursor measurement on the XY display for safe operating area analysis.

Syntax :ANALysis:PANalyze:SOA:CURSOr?

:ANALysis:PANalyze:SOA:CURSOr:X<x>?

Function Queries all settings related to the horizontal cursors on the XY display for safe operating area analysis.

Syntax :ANALysis:PANalyze:SOA:CURSOr:X<x>?
<x> = 1 to 2

:ANALysis:PANalyze:SOA:CURSOr:X<x>:POSition

Function Sets or queries the horizontal cursor positions on the XY display for safe operating area analysis.

Syntax :ANALysis:PANalyze:SOA:CURSOr:X<x>:
POSition {<NRf>}
:ANALysis:PANalyze:SOA:CURSOr:X<x>:
POSition?
<x> = 1 to 2
<NRf> = -4 to 4 (div)

Example :ANALYSIS:PANALYZE:SOA:Cursors:X1:
POSITION 1
:ANALYSIS:PANALYZE:SOA:Cursors:X1:
POSITION?
-> :ANALYSIS:PANALYZE:SOA:Cursors:X:
POSITION 1.000E+00

:ANALysis:PANalyze:SOA:CURSOr:X<x>:VALue?

Function Queries the horizontal cursor voltages on the XY display for safe operating area analysis.

Syntax :ANALysis:PANalyze:SOA:CURSOr:X<x>:
VALue?
<x> = 1 to 2

Example :ANALYSIS:PANALYZE:SOA:Cursors:X1:
VALUE?
-> :ANALYSIS:PANALYZE:SOA:Cursors:X1:
VALUE 1.000E+00

:ANALysis:PANalyze:SOA:CURSOr:Y<y>?

Function Queries all settings related to the vertical cursors on the XY display for safe operating area analysis.

Syntax :ANALysis:PANalyze:SOA:CURSOr:Y<y>?
<y> = 1 to 2

:ANALysis:PANalyze:SOA:CURSOr:Y<y>:POSition

Function Sets or queries the vertical cursor positions on the XY display for safe operating area analysis.

Syntax :ANALysis:PANalyze:SOA:CURSOr:Y<y>:
POSition {<NRf>}
:ANALysis:PANalyze:SOA:CURSOr:Y<y>:
POSition?
<y> = 1 to 2
<NRf> = -4 to 4 (div)

Example :ANALYSIS:PANALYZE:SOA:Cursors:Y1:
POSITION 1
:ANALYSIS:PANALYZE:SOA:Cursors:Y1:
POSITION?
-> :ANALYSIS:PANALYZE:SOA:Cursors:Y:
POSITION 1.000E+00

5.3 ANALysis Group

:ANALysis:PANalyze:SOA:CURSor:Y<x>:VALue?

Function Queries the vertical cursor voltages on the XY display for safe operating area analysis.

Syntax :ANALysis:PANalyze:SOA:CURSor:Y<x>:VALue?
<x> = 1 to 2

Example :ANALYSIS:PANALYZE:SOA:CURSOR:Y1:VALUE?
-> :ANALYSIS:PANALYZE:SOA:CURSOR:Y1:VALUE 1.000E+00

:ANALysis:PANalyze:SOA:MODE

Function Sets or queries the automated measurement mode on the XY display for safe operating area analysis.

Syntax :ANALysis:PANalyze:SOA:MODE {CURSor|OFF}

Example :ANALYSIS:PANALYZE:SOA:MODE? :ANALYSIS:PANALYZE:SOA:MODE CURSOR :ANALYSIS:PANALYZE:SOA:MODE? -> :ANALYSIS:PANALYZE:SOA:MODE CURSOR

:ANALysis:PANalyze:SOA:TRANge (Time Range)

Function Sets or queries the VT waveform range to show on the XY display for safe operating area analysis.

Syntax :ANALysis:PANalyze:SOA:TRANge {<NRf>,<NRf>}
:ANALysis:PANalyze:SOA:TRANge? <NRf> = -5 to 5 (div), (10 div/display record length) steps

Example :ANALYSIS:PANALYZE:SOA:TRANGE -4,4 :ANALYSIS:PANALYZE:SOA:TRANGE? -> :ANALYSIS:PANALYZE:SOA:TRANGE -4.00,4.00

:ANALysis:PANalyze:SOA:VTDisplay

Function Sets or queries the on/off state of the VT waveform display on the XY display for safe operating area analysis.

Syntax :ANALysis:PANalyze:SOA:VTDisplay {<Boolean>}
:ANALysis:PANalyze:SOA:VTDisplay?

Example :ANALYSIS:PANALYZE:SOA:VTDISPLAY ON :ANALYSIS:PANALYZE:SOA:VTDISPLAY? -> :ANALYSIS:PANALYZE:SOA:VTDISPLAY 1

:ANALysis:PANalyze:SWLoss?

Function Queries all switching loss settings.

Syntax :ANALysis:PANalyze:SWLoss?

Description Use the following command to set the reference levels (e.g., distal, mesial, proximal) for determining the total loss.

:MEASure:{CHANnel<x>|MATH<x>} : DPROximal?
:MEASure:{CHANnel<x>|MATH<x>} : DPROximal:MODE
:MEASure:{CHANnel<x>|MATH<x>} : DPROximal:PERCent
:MEASure:{CHANnel<x>|MATH<x>} : DPROximal:UNIT
:MEASure:{CHANnel<x>|MATH<x>} :METHOD

:ANALysis:PANalyze:SWLoss:CCALc

Function Sets or queries the total loss equation.

Syntax :ANALysis:PANalyze:SWLoss:CCALc{RDS|VCE|WAVEform}

Example :ANALYSIS:PANALYZE:SWLOSS:CCALC RDS :ANALYSIS:PANALYZE:SWLOSS:CCALC? -> :ANALYSIS:PANALYZE:SWLOSS:CCALC RDS

:ANALysis:PANalyze:SWLoss:DPROximal?

Function Queries all distal, mesial, and proximal settings.

Syntax :ANALysis:PANalyze:SWLoss:DPROximal?

:ANALysis:PANalyze:SWLoss:DPROximal:MODE

Function Sets or queries the distal, mesial, and proximal point mode setting.

Syntax :ANALysis:PANalyze:SWLoss:DPROximal:MODE {PERCent|UNIT}
:ANALysis:PANalyze:SWLoss:DPROximal:MODE?

Example :ANALYSIS:PANALYZE:SWLOSS:DPROXIMAL:MODE PERCENT :ANALYSIS:PANALYZE:SWLOSS:DPROXIMAL:MODE? -> :ANALYSIS:PANALYZE:SWLOSS:DPROXIMAL:MODE PERCENT

:ANALysis:PANalyze:SWLoss:DPROximal:PERCent

Function Sets or queries the distal, mesial, and proximal points as percentages.

Syntax :ANALysis:PANalyze:SWLoss:DPROximal:PERCent {<NRf>,<NRf>,<NRf>}
:ANALysis:PANalyze:SWLoss:DPROximal:PERCent?
<NRf>, <NRf>, <NRf> = 0 to 100 (% in steps of 1)

Example :ANALYSIS:PANALYZE:SWLOSS:DPROXIMAL:PERCENT 10,50,90
:ANALYSIS:PANALYZE:SWLOSS:DPROXIMAL:PERCENT?
-> :ANALYSIS:PANALYZE:SWLOSS:DPROXIMAL:PERCENT 10,50,90

:ANALysis:PANalyze:SWLoss:DPROximal:UNIT

Function Sets or queries the distal, mesial, and proximal points as voltages.

Syntax :ANALysis:PANalyze:SWLoss:DPROximal:UNIT {<NRf>,<NRf>,<NRf>}
:ANALysis:PANalyze:SWLoss:DPROximal:UNIT?
<NRf>, <NRf>, <NRf> = See the Features Guide for this information.

Example :ANALYSIS:PANALYZE:SWLOSS:DPROXIMAL:UNIT -1,0,1
:ANALYSIS:PANALYZE:SWLOSS:DPROXIMAL:UNIT?
-> :ANALYSIS:PANALYZE:SWLOSS:DPROXIMAL:UNIT -1.0000000E+00,0.0000000E+00,1.0000000E+00

:ANALysis:PANalyze:SWLoss:ILEVel

Function Sets or queries the current level for determining the loss zero period of total loss.

Syntax :ANALysis:PANalyze:SWLoss:ILEVel {<NRf>|<Current>}
:ANALysis:PANalyze:SWLoss:ILEVel?
<NRf>, <current> = See the Features Guide for this information.

Example :ANALYSIS:PANALYZE:SWLOSS:ILEVEL 1
:ANALYSIS:PANALYZE:SWLOSS:ILEVEL?
-> :ANALYSIS:PANALYZE:SWLOSS:ILEVEL 1.0000000E+00

:ANALysis:PANalyze:SWLoss:INDicator

Function Sets or queries the total loss measurement location indicator.

Syntax :ANALysis:PANalyze:SWLoss:INDicator {<Boolean>}
:ANALysis:PANalyze:SWLoss:INDicator?

Example :ANALYSIS:PANALYZE:SWLOSS:INDICATOR OFF
:ANALYSIS:PANALYZE:SWLOSS:INDICATOR?
-> :ANALYSIS:PANALYZE:SWLOSS:INDICATOR OFF

:ANALysis:PANalyze:SWLoss:MATH

Function Sets or queries the power waveform display on/off state.

Syntax :ANALysis:PANalyze:SWLoss:MATH {OFF|Power}
:ANALysis:PANalyze:SWLoss:MATH?

Example :ANALYSIS:PANALYZE:SWLOSS:MATH OFF
:ANALYSIS:PANALYZE:SWLOSS:MATH?
-> :ANALYSIS:PANALYZE:SWLOSS:MATH OFF

:ANALysis:PANalyze:SWLoss:MEASure?

Function Queries all settings related to the automated measurement of power supply parameters.

Syntax :ANALysis:PANalyze:SWLoss:MEASure?

:ANALysis:PANalyze:SWLoss:MEASure:<Parameter>?

Function Queries all power supply analysis parameters.

Syntax :ANALysis:PANalyze:SWLoss:MEASure:<Parameter>?
<Parameter> = {CCount|PON|PTOFF|PTON|PTOTail|WHON|WHTOFF|WHTON|WHTOTail}

:ANALysis:PANalyze:SWLoss:MEASure:<Parameter>:COUNT?

Function Queries the continuous statistical processing count of the power supply analysis parameter.

Syntax :ANALysis:PANalyze:SWLoss:MEASure:<Parameter>:COUNT?
<Parameter> = {CCount|PON|PTOFF|PTON|PTOTail|WHON|WHTOFF|WHTON|WHTOTail}

Example :ANALYSIS:PANALYZE:SWLOSS:MEASURE:P:COUNT?
-> :ANALYSIS:PANALYZE:SWLOSS:MEASURE:P:COUNT 100

5.3 ANALysis Group

:ANALysis:PANalyze:SWLoss:MEASure:<Parameter>:{MAXimum|MEAN|MINimum|SDEVIation}?

Function Queries a statistical value of a power supply analysis parameter.

Syntax :ANALysis:PANalyze:SWLoss:MEASure:<Parameter>:{MAXimum|MEAN|MINimum|SDEVIation}?
<Parameter> = {CCOunt|PON|PTOf|PTON|PTOTal|WHON|WHTOf|WHTON|WHTOTal}

Example :ANALYSIS:PANALYZE:SWLOSS:MEASURE:P:MAXIMUM?
-> :ANALYSIS:PANALYZE:SWLOSS:MEASURE:P:MAXIMUM 1.000E+00

Description If the statistical value cannot be obtained, the instrument returns "NAN" (not a number).

:ANALysis:PANalyze:SWLoss:MEASure:<Parameter>:STATe

Function Sets or queries the on/off state of the power supply analysis parameter.

Syntax :ANALysis:PANalyze:SWLoss:MEASure:<Parameter>:STATe {<Boolean>}
:ANALysis:PANalyze:SWLoss:MEASure:<Parameter>:STATe?
<Parameter> = {CCOunt|PON|PTOf|PTON|PTOTal|WHON|WHTOf|WHTON|WHTOTal}

Example :ANALYSIS:PANALYZE:SWLOSS:MEASURE:P:STATE ON
:ANALYSIS:PANALYZE:SWLOSS:MEASURE:P:STATE?
-> :ANALYSIS:PANALYZE:SWLOSS:MEASURE:P:STATE 1

:ANALysis:PANalyze:SWLoss:MEASure:<Parameter>:VALue?

Function Queries the automatically measured value of a power supply parameter.

Syntax :ANALysis:PANalyze:SWLoss:MEASure:<Parameter>:VALue? [{<NRf>}]
<NRf> = See the Features Guide for this information.
<Parameter> = {CCOunt|PON|PTOf|PTON|PTOTal|WHON|WHTOf|WHTON|WHTOTal}

Example :ANALYSIS:PANALYZE:SWLOSS:MEASURE:P:VALUE?
-> :ANALYSIS:PANALYZE:SWLOSS:MEASURE:P:VALUE 10.0000E+03

Description

- If the value is immeasurable, the instrument returns "NAN" (not a number).
- The <NRf> is used to specify which iteration of automated measurement to query the measured value from.
- If <NRf> is set to 1, the oldest measured value in the automated measurement memory is queried.
- If a measured value does not exist at the specified iteration, the instrument returns "NAN" (not a number).
- If <NRf> is omitted, the most recent measured value is queried.
- If cyclic statistical processing for automated measurement is being executed and <NRf> is specified, the measured values over a cycle in iteration <NRf> from the left of the display on the displayed waveform is queried. If <NRf> is omitted, the measured values over the last cycle on the displayed waveform is queried.

:ANALysis:PANalyze:SWLoss:MEASure:CONTinuous?

Function Queries all settings related to the normal statistical processing of automatically measured total loss.

:ANALysis:PANalyze:SWLoss:MEASure:CONTinuous:REStart

Function Restarts the normal statistical processing of automatically measured total loss.

Syntax :ANALysis:PANalyze:SWLoss:MEASure:CONTinuous:REStart

Example :ANALYSIS:PANALYZE:SWLOSS:MEASURE:CONTINUOUS:RESTART

:ANALysis:PANalyze:SWLoss:MEASure:CONTInuous:TLCHange

Function Sets or queries whether the normal statistical processing of automatically measured total loss is to be restarted.

Syntax :ANALysis:PANalyze:SWLoss:MEASure:CONTInuous:TLCHange {REStart|IGNore}
:ANALysis:PANalyze:SWLoss:MEASure:CONTInuous:TLCHange?

Example :ANALYSIS:PANALYZE:SWLOSS:MEASURE:CONTINUOUS:TLCHANGE RESTART
:ANALYSIS:PANALYZE:SWLOSS:MEASURE:CONTINUOUS:TLCHANGE?
-> :ANALYSIS:PANALYZE:SWLOSS:MEASURE:CONTINUOUS:TLCHANGE RESTART

:ANALysis:PANalyze:SWLoss:MEASure:CYCLe:ABORT

Function Aborts the execution of the cyclic statistical processing of automatically measured total loss.

Syntax :ANALysis:PANalyze:SWLoss:MEASure:CYCLe:ABORT

Example :ANALYSIS:PANALYZE:SWLOSS:MEASURE:CYCLE:ABORT

:ANALysis:PANalyze:SWLoss:MEASure:CYCLe:EXECute

Function Executes the cyclic statistical processing of automatically measured total loss.

Syntax :ANALysis:PANalyze:SWLoss:MEASure:CYCLe:EXECute

Example :ANALYSIS:PANALYZE:SWLOSS:MEASURE:CYCLE:EXECUTE

:ANALysis:PANalyze:SWLoss:MEASure:HISTory:ABORT

Function Aborts the execution of the statistical processing of the history waveform of automatically measured total loss.

Syntax :ANALysis:PANalyze:SWLoss:MEASure:HISTory:ABORT

Example :ANALYSIS:PANALYZE:SWLOSS:MEASURE:HISTORY:ABORT

:ANALysis:PANalyze:SWLoss:MEASure:HISTory:EXECute

Function Executes the statistical processing of the history waveform of automatically measured total loss.

Syntax :ANALysis:PANalyze:SWLoss:MEASure:HISTory:EXECute

Example :ANALYSIS:PANALYZE:SWLOSS:MEASURE:HISTORY:EXECUTE

:ANALysis:PANalyze:SWLoss:MEASure:MODE

Function Sets or queries the statistical processing type of automatically measured total loss.

Syntax :ANALysis:PANalyze:SWLoss:MEASure:MODE {OFF|CONTInuous|CYCLe|HISTory}
:ANALysis:PANalyze:SWLoss:MEASure:MODE?

Example :ANALYSIS:PANALYZE:SWLOSS:MEASURE:MODE OFF
:ANALYSIS:PANALYZE:SWLOSS:MEASURE:MODE?
-> :ANALYSIS:PANALYZE:SWLOSS:MEASURE:MODE OFF

:ANALysis:PANalyze:SWLoss:MEASure:ZLINKage

Function Sets or queries the linkage (on/off) between the result numbers and zoom positions of the results of automatically measured total loss.

Syntax :ANALysis:PANalyze:SWLoss:MEASure:ZLINKage {<Boolean>}
:ANALysis:PANalyze:SWLoss:MEASure:ZLINKage?

Example :ANALYSIS:PANALYZE:SWLOSS:MEASURE:ZLINKAGE ON
:ANALYSIS:PANALYZE:SWLOSS:MEASURE:ZLINKAGE?
-> :ANALYSIS:PANALYZE:SWLOSS:MEASURE:ZLINKAGE ON

:ANALysis:PANalyze:SWLoss:METHod

Function Sets or queries the calculation method for high and low points.

Syntax :ANALysis:PANalyze:SWLoss:METHod {AUTO|MAXimum|HISTogram}
:ANALysis:PANalyze:SWLoss:METHod?

Example :ANALYSIS:PANALYZE:SWLOSS:METHOD AUTO
:ANALYSIS:PANALYZE:SWLOSS:METHOD?
-> :ANALYSIS:PANALYZE:SWLOSS:METHOD AUTO

:ANALysis:PANalyze:SWLoss:RANGE

Function Sets or queries the measurement source window.

Syntax :ANALysis:PANalyze:SWLoss:RANGE {MAIN|Z1|Z2}
:ANALysis:PANalyze:SWLoss:RANGE?

Example :ANALYSIS:PANALYZE:SWLOSS:RANGE MAIN
:ANALYSIS:PANALYZE:SWLOSS:RANGE?
-> :ANALYSIS:PANALYZE:SWLOSS:RANGE MAIN

5.3 ANALysis Group

:ANALysis:PANalyze:SWLoss:RDS

Function Sets or queries the total loss on-resistance.

Syntax :ANALysis:PANalyze:SWLoss:
RDS {<NRf>}
:ANALysis:PANalyze:SWLoss:RDS?
<NRf> = 0 to 100 (in 1 mΩ steps)

Example :ANALYSIS:PANALYZE:SWLOSS:RDS 1
:ANALYSIS:PANALYZE:SWLOSS:RDS?
-> :ANALYSIS:PANALYZE:SWLOSS:
RDS 1.000E+00

:ANALysis:PANalyze:SWLoss:SCALE?

Function Queries all scaling settings.

Syntax :ANALysis:PANalyze:SWLoss:SCALE?

:ANALysis:PANalyze:SWLoss:SCALE:CENTer

Function Sets or queries the center value for manual scaling.

Syntax :ANALysis:PANalyze:SWLoss:SCALE:
CENTer {<NRf>}
:ANALysis:PANalyze:SWLoss:SCALE:
CENTer?

Example :ANALYSIS:PANALYZE:SWLOSS:SCALE:
CENTER 1
:ANALYSIS:PANALYZE:SWLOSS:SCALE:
CENTER?
-> :ANALYSIS:PANALYZE:SWLOSS:SCALE:
CENTER 1.00000E+00

:ANALysis:PANalyze:SWLoss:SCALE:MODE

Function Sets or queries the scale mode.

Syntax :ANALysis:PANalyze:SWLoss:SCALE:
MODE {AUTO|MANual}
:ANALysis:PANalyze:SWLoss:SCALE:
MODE?

Example :ANALYSIS:PANALYZE:SWLOSS:SCALE:
MODE AUTO
:ANALYSIS:PANALYZE:SWLOSS:SCALE:
MODE?
-> :ANALYSIS:PANALYZE:SWLOSS:SCALE:
MODE AUTO

:ANALysis:PANalyze:SWLoss:SCALE:SENSitivity

Function Sets or queries the sensitivity for manual scaling.

Syntax :ANALysis:PANalyze:SWLoss:SCALE:
SENSitivity {<NRf>}
:ANALysis:PANalyze:SWLoss:SCALE:
SENSitivity?

Example :ANALYSIS:PANALYZE:SWLOSS:SCALE:
SENSITIVITY 10
:ANALYSIS:PANALYZE:SWLOSS:SCALE:
SENSITIVITY?
-> :ANALYSIS:PANALYZE:SWLOSS:SCALE:
SENSITIVITY 10.0000E+00

:ANALysis:PANalyze:SWLoss:TRANGE (Time Range)

Function Sets or queries the measurement time period.

Syntax :ANALysis:PANalyze:SWLoss:
TRANGE {<NRf>,<NRf>}
:ANALysis:PANalyze:SWLoss:TRANGE?
<NRf> = -5 to 5 (div), (10 div/display record
length) steps

Example :ANALYSIS:PANALYZE:SWLOSS:TRANGE -4,
4
:ANALYSIS:PANALYZE:SWLOSS:TRANGE?
-> :ANALYSIS:PANALYZE:SWLOSS:
TRANGE -4.00E+00,4.00E+00

:ANALysis:PANalyze:SWLoss:ULEVel

Function Sets or queries the voltage level for determining the loss calculation period of total loss.

Syntax :ANALysis:PANalyze:SWLoss:
ULEVel {<NRf>|<Voltage>}
:ANALysis:PANalyze:SWLoss:ULEVel?
<NRf>, <voltage> = See the Features Guide for
this information.

Example :ANALYSIS:PANALYZE:SWLOSS:ULEVEL 1V
:ANALYSIS:PANALYZE:SWLOSS:ULEVEL?
-> :ANALYSIS:PANALYZE:SWLOSS:
ULEVEL 1.0000000E+00

:ANALysis:PANalyze:SWLoss:UNIT

Function Sets or queries the watt-hour unit.

Syntax :ANALysis:PANalyze:SWLoss:UNIT {J|
WH}
:ANALysis:PANalyze:SWLoss:UNIT?

Example :ANALYSIS:PANALYZE:SWLOSS:UNIT WH
:ANALYSIS:PANALYZE:SWLOSS:UNIT?
-> :ANALYSIS:PANALYZE:SWLOSS:UNIT WH

:ANALysis:PANalyze:SWLoss:VCE

Function Sets or queries the collector-emitter saturation voltage of total loss.

Syntax :ANALysis:PANalyze:SWLoss:
VCE {<NRf>|<Voltage>}
:ANALysis:PANalyze:SWLoss:VCE?
<NRf>, <Voltage> = 0 V to 50 V (100 mV steps)

Example :ANALYSIS:PANALYZE:SWLOSS:VCE 1V
:ANALYSIS:PANALYZE:SWLOSS:VCE?
-> :ANALYSIS:PANALYZE:SWLOSS:
VCE 1.0000000E+00

:ANALysis:PANalyze:TYPE

Function Sets or queries the power supply analysis type.

Syntax :ANALysis:PANalyze:TYPE {HARMonics|
I2T|OFF|SOA|SWLoss}
:ANALysis:PANalyze:TYPE?
Example :ANALYSIS:PANALYZE:TYPE HARMONICS
:ANALYSIS:PANALYZE:TYPE?
-> :ANALYSIS:PANALYZE:TYPE HARMONICS

:ANALysis:PMEASURE<x>?

Function Queries all power measurement settings.

Syntax :ANALysis:PMEASURE<x>?
<x> = 1 to 2

:ANALysis:PMEASURE<x>:CYCMode

Function Sets or queries power measurement cycle mode.

Syntax :ANALysis:PMEASURE:CYCMode {OFF|
NCYCLe}
:ANALysis:PMEASURE:CYCMode?
<x> = 1 to 2

Example :ANALYSIS:PMEASURE:CYCMODE OFF
:ANALYSIS:PMEASURE:CYCMODE?
-> :ANALYSIS:PMEASURE:CYCMODE OFF

:ANALysis:PMEASURE<x>:CYCTrace

Function Sets or queries the source waveform of power measurement cycle mode.

Syntax :ANALysis:PMEASURE:CYCTrace {<NRf>|
MATH<y>}
:ANALysis:PMEASURE:CYCTrace?
<x> = 1 to 2
<NRf> = 1 to 4
<y> = 1 to 4

Example :ANALYSIS:PMEASURE:CYCTRACE 1
:ANALYSIS:PMEASURE:CYCTRACE?
-> :ANALYSIS:PMEASURE:CYCTRACE 1

:ANALysis:PMEASURE<x>:IDPROximal?

Function Queries all distal, mesial, and proximal settings.

Syntax :ANALysis:PMEASURE<x>:IDPROximal?
<x> = 1 to 2

:ANALysis:PMEASURE<x>:IDPROximal:MODE

Function Sets or queries the distal, mesial, and proximal point mode setting.

Syntax :ANALysis:PMEASURE<x>:IDPROximal:
MODE {PERCent|UNIT}
:ANALysis:PMEASURE<x>:IDPROximal:
MODE?
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:IDPROXIMAL:
MODE PERCENT
:ANALYSIS:PMEASURE1:IDPROXIMAL:MODE?
-> :ANALYSIS:PMEASURE1:IDPROXIMAL:
MODE PERCENT

:ANALysis:PMEASURE<x>:IDPROximal:PERCent

Function Sets or queries the distal, mesial, and proximal points as percentages.

Syntax :ANALysis:PMEASURE<x>:IDPROximal:
PERCent {<NRf>,<NRf>,<NRf>}
:ANALysis:PMEASURE<x>:IDPROximal:
PERCent?
<x> = 1 to 2
<NRf>, <NRf>, <NRf> = 0 to 100 (% in steps of 1)

Example :ANALYSIS:PMEASURE1:IDPROXIMAL:
PERCENT 10,50,90
:ANALYSIS:PMEASURE1:IDPROXIMAL:
PERCENT?
-> :ANALYSIS:PMEASURE1:IDPROXIMAL:
PERCENT 10,50,90

:ANALysis:PMEASURE<x>:IDPROximal:UNIT

Function Sets or queries the distal, mesial, and proximal points as voltages.

Syntax :ANALysis:PMEASURE<x>:IDPROximal:
UNIT {<NRf>,<NRf>,<NRf>}
:ANALysis:PMEASURE<x>:IDPROximal:
UNIT?
<x> = 1 to 2
<NRf>, <NRf>, <NRf> = See the Features Guide for this information.

Example :ANALYSIS:PMEASURE1:IDPROXIMAL:
UNIT -1,0,1
:ANALYSIS:PMEASURE1:IDPROXIMAL:UNIT?
-> :ANALYSIS:PMEASURE1:IDPROXIMAL:
UNIT -1.0000000E+00,0.0000000E+00,
1.0000000E+00

5.3 ANALysis Group

:ANALysis:PMEASURE<x>:IMETHOD

Function Sets or queries the high and low points.

Syntax :ANALysis:PMEASURE<x>:IMETHOD {AUTO|
MAXimum|HISTogram}
:ANALysis:PMEASURE<x>:IMETHOD?
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:IMETHOD AUTO
:ANALYSIS:PMEASURE1:IMETHOD?
-> :ANALYSIS:PMEASURE1:IMETHOD AUTO

:ANALysis:PMEASURE<x>:INDICATOR

Function Sets or queries the measurement location indicator.

Syntax :ANALysis:PMEASURE<x>:
INDicator <Parameter>
:ANALysis:PMEASURE<x>:INDicator?
<x> = 1 to 2
<Parameter> = {AH|AHABs|AHN|AHP|IAC|Iavg
freq|IDC|IMN|INPeak|IPPeak|IPTOpeak|IRMN|IR
MS|LAMBda|P|Q|S|UAC|UAVGfreq|UDC|UMN|U
NPeak|UPPeak|UPTOpeak|URMN|URMS|WH|W
HABs|WHN|WHP|Z}

Example :ANALYSIS:PMEASURE1:INDICATOR AH
:ANALYSIS:PMEASURE1:INDICATOR?
-> :ANALYSIS:PMEASURE1:INDICATOR AH

:ANALysis:PMEASURE<x>:MEASURE?

Function Queries all settings related to the automated measurement of power measurement parameters.

Syntax :ANALysis:PMEASURE<x>:MEASURE?
<x> = 1 to 2

:ANALysis:PMEASURE<x>:MEASURE:<Parameter>?

Function Queries the power measurement parameter settings.

Syntax :ANALysis:PMEASURE<x>:MEASURE:
<Parameter>?
<x> = 1 to 2
<Parameter> = {AH|AHABs|AHN|AHP|IAC|Iavg
freq|IDC|IMN|INPeak|IPPeak|IPTOpeak|IRMN|IR
MS|LAMBda|P|Q|S|UAC|UAVGfreq|UDC|UMN|U
NPeak|UPPeak|UPTOpeak|URMN|URMS|WH|W
HABs|WHN|WHP|Z}

Example :ANALYSIS:PMEASURE1:MEASURE:AH?

:ANALysis:PMEASURE<x>:MEASURE:<Parameter>:COUNT?

Function Queries the normal statistical processing count of the power measurement parameter.

Syntax :ANALysis:PMEASURE<x>:MEASURE:
<Parameter>:COUNT?
<x> = 1 to 2
<Parameter> = {AH|AHABs|AHN|AHP|IAC|Iavg
freq|IDC|IMN|INPeak|IPPeak|IPTOpeak|IRMN|IR
MS|LAMBda|P|Q|S|UAC|UAVGfreq|UDC|UMN|U
NPeak|UPPeak|UPTOpeak|URMN|URMS|WH|W
HABs|WHN|WHP|Z}

Example :ANALYSIS:PMEASURE1:MEASURE:AH:
COUNT?

:ANALysis:PMEASURE<x>:MEASURE:<Parameter>:{MAXimum|MEAN|MINimum|SDEVIATION}?

Function Queries the statistical value of a power measurement parameter.

Syntax :ANALysis:PMEASURE<x>:MEASURE:
<Parameter>:{MAXimum|MEAN|MINimum|
SDEVIATION}?
<x> = 1 to 2
<Parameter> = {AH|AHABs|AHN|AHP|IAC|Iavg
freq|IDC|IMN|INPeak|IPPeak|IPTOpeak|IRMN|IR
MS|LAMBda|P|Q|S|UAC|UAVGfreq|UDC|UMN|U
NPeak|UPPeak|UPTOpeak|URMN|URMS|WH|W
HABs|WHN|WHP|Z}

Example :ANALYSIS:PMEASURE1:MEASURE:AH:
MAXIMUM?

:ANALysis:PMEASURE<x>:MEASURE:<Parameter>:STATE

Function Sets or queries the on/off state of a power measurement parameter.

Syntax :ANALysis:PMEASURE<x>:MEASURE:
<Parameter>:STATE {<Boolean>}
:ANALysis:PMEASURE<x>:MEASURE:
<Parameter>:STATE?
<x> = 1 to 2
<Parameter> = {AH|AHABs|AHN|AHP|IAC|Iavg
freq|IDC|IMN|INPeak|IPPeak|IPTOpeak|IRMN|IR
MS|LAMBda|P|Q|S|UAC|UAVGfreq|UDC|UMN|U
NPeak|UPPeak|UPTOpeak|URMN|URMS|WH|W
HABs|WHN|WHP|Z}

Example :ANALYSIS:PMEASURE1:MEASURE:AH:
STATE ON
:ANALYSIS:PMEASURE1:MEASURE:AH:
STATE?
-> :ANALYSIS:PMEASURE1:MEASURE:AH:
STATE 1

:ANALysis:PMEASURE<x>:MEASURE:<Parameter>:VALUE?

Function	Queries the automatically measured power measurement parameter value.
Syntax	:ANALysis:PMEASURE<x>:MEASURE: <Parameter>:VALUE? <x> = 1, 2 <Parameter> = {AH AHABs AHN AHP IAC IAVG freq IDC IMN INPeak IPPeak IPTopPeak IRMN IR MS LAMBda P Q S UAC UAVGfreq UDC UMN U NPeak UPPeak UPTopPeak URMN URMS WH W HABs WHN WHP Z}
Example	:ANALYSIS:PMEASURE1:MEASURE:AH: VALUE?

:ANALysis:PMEASURE<x>:MEASURE:ALL

Function	Collectively turns on or off the power measurement parameters.
Syntax	:ANALysis:PMEASURE<x>:MEASURE: ALL {<Boolean>} <x> = 1 to 2
Example	:ANALYSIS:PMEASURE1:MEASURE:ALL ON

:ANALysis:PMEASURE<x>:MEASURE:CONTInuous

Function	Queries all settings related to the normal statistical processing of automated power measurement.
Syntax	:ANALysis:PMEASURE<x>:MEASURE: CONTInuous? <x> = 1 to 2

:ANALysis:PMEASURE<x>:MEASURE:CONTInuous:REStart

Function	Restarts the normal statistical processing of automated power measurement.
Syntax	:ANALysis:PMEASURE<x>:MEASURE: CONTInuous:REStart <x> = 1 to 2
Example	:ANALYSIS:PMEASURE1:MEASURE: CONTINUOUS:RESTART

:ANALysis:PMEASURE<x>:MEASURE:CONTInuous:TLCHange

Function	Sets or queries whether the normal statistical processing of automated power measurement is to be restarted.
Syntax	:ANALysis:PMEASURE<x>:MEASURE: CONTInuous:TLCHange {REStart IGNore} :ANALysis:PMEASURE<x>:MEASURE: CONTInuous:TLCHange? <x> = 1 to 2
Example	:ANALYSIS:PMEASURE1:MEASURE: CONTINUOUS:TLCHANGE RESTART :ANALYSIS:PMEASURE1:MEASURE: CONTINUOUS:TLCHANGE -> :ANALYSIS:PMEASURE1:MEASURE: CONTINUOUS:TLCHANGE RESTART

:ANALysis:PMEASURE<x>:MEASURE:CYCLE?

Function	Queries all settings related to the cyclic statistical processing of automated power measurement.
Syntax	:ANALysis:PMEASURE<x>:MEASURE:CYCLE? <x> = 1 to 2

:ANALysis:PMEASURE<x>:MEASURE:CYCLE:ABORT

Function	Aborts the execution of the cyclic statistical processing of automated power measurement.
Syntax	:ANALysis:PMEASURE<x>:MEASURE:CYCLE: ABORT <x> = 1 to 2
Example	:ANALYSIS:PMEASURE1:MEASURE:CYCLE: ABORT

:ANALysis:PMEASURE<x>:MEASURE:CYCLE:EXECute

Function	Executes the cyclic statistical processing of automated power measurement.
Syntax	:ANALysis:PMEASURE<x>:MEASURE:CYCLE: EXECute <x> = 1 to 2
Example	:ANALYSIS:PMEASURE1:MEASURE:CYCLE: EXECUTE

5.3 ANALysis Group

:ANALysis:PMEASURE<x>:MEASURE:CYCLE:TRACE

Function Sets or queries the source waveform of the cyclic statistical processing of automated power measurement.

Syntax :ANALysis:PMEASURE<x>:MEASURE:CYCLE:TRACE {<NRf>|MATH<y>}
:ANALysis:PMEASURE<x>:MEASURE:CYCLE:TRACE?
<x> = 1 to 2
<NRf> = 1 to 4
<y> = 1 to 4

Example :ANALYSIS:PMEASURE1:MEASURE:CYCLE:TRACE 1
:ANALYSIS:PMEASURE1:MEASURE:CYCLE:TRACE?
-> :ANALYSIS:PMEASURE1:MEASURE:CYCLE:TRACE 1

:ANALysis:PMEASURE<x>:MEASURE:HISTORY:ABORT

Function Aborts the statistical processing of the history waveform of automated power measurement.

Syntax :ANALysis:PMEASURE<x>:MEASURE:HISTORY:ABORT
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:MEASURE:HISTORY:ABORT

:ANALysis:PMEASURE<x>:MEASURE:HISTORY:EXECUTE

Function Executes the statistical processing of the history waveform of automated power measurement.

Syntax :ANALysis:PMEASURE<x>:MEASURE:HISTORY:EXECUTE
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:MEASURE:HISTORY:EXECUTE

:ANALysis:PMEASURE<x>:MEASURE:MODE

Function Sets or queries the statistical processing type of automated power measurement.

Syntax :ANALysis:PMEASURE<x>:MEASURE:MODE {OFF|CONTINUOUS|CYCLE|HISTORY}
:ANALysis:PMEASURE<x>:MEASURE:MODE?
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:MEASURE:MODE OFF
:ANALYSIS:PMEASURE1:MEASURE:MODE?
-> :ANALYSIS:PMEASURE1:MEASURE:MODE OFF

:ANALysis:PMEASURE<x>:MEASURE:ZLINKage

Function Sets or queries the linkage (on/off) between the result numbers and zoom positions of the results of automated power measurement.

Syntax :ANALysis:PMEASURE<x>:MEASURE:ZLINKage {<Boolean>}
:ANALysis:PMEASURE<x>:MEASURE:ZLINKage?
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:MEASURE:ZLINKAGE ON
:ANALYSIS:PMEASURE1:MEASURE:ZLINKAGE?
-> :ANALYSIS:PMEASURE1:MEASURE:ZLINKAGE ON

:ANALysis:PMEASURE<x>:MODE

Function Sets or queries the on/off state of power measurement.

Syntax :ANALysis:PMEASURE<x>:MODE {<Boolean>}
:ANALysis:PMEASURE<x>:MODE?
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:MODE ON
:ANALYSIS:PMEASURE1:MODE?
-> :ANALYSIS:PMEASURE1:MODE 1

:ANALysis:PMEASURE<x>:RANGE

Function Sets or queries the measurement source window.

Syntax :ANALysis:PMEASURE<x>:RANGE {MAIN|Z1|Z2}
:ANALysis:PMEASURE<x>:RANGE?
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:RANGE MAIN
:ANALYSIS:PMEASURE1:RANGE?
-> :ANALYSIS:PMEASURE1:RANGE MAIN

:ANALysis:PMEASURE<x>:SETup?

Function Queries all power measurement input settings.

Syntax :ANALysis:PMEASURE<x>:SETup?
<x> = 1 to 2

:ANALysis:PMEASURE<x>:SETup:ADESkew

Function Executes auto deskewing for power measurement.

Syntax :ANALysis:PMEASURE<x>:SETup:ADESkew
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:SETUP:ADESKEW

:ANALysis:PMEasure<x>:SETup:I?

Function Queries all current input channel settings for power measurement.

Syntax :ANALysis:PMEasure<x>:SETup:I?
<x> = 1 to 2

:ANALysis:PMEasure<x>:SETup:I:DESKew

Function Sets or queries deskewing of the current input channel for power measurement.

Syntax :ANALysis:PMEasure<x>:SETup:I:
DESKew {<Time>}
:ANALysis:PMEasure<x>:SETup:I:
DESKew?
<x> = 1 to 2
<Time> = -1000.00 ns to 1000.00 ns (10 ps steps)

Example :ANALYSIS:PMEASURE1:SETUP:I:
DESKew 1NS
:ANALYSIS:PMEASURE1:SETUP:I:DESKew?
-> :ANALYSIS:PMEASURE1:SETUP:I:
DESKew 1.000E-09

:ANALysis:PMEasure<x>:SETup:I:PROBe

Function Sets or queries the current-to-voltage conversion ratio settings of the probe connected to the current input channel for power measurement.

Syntax :ANALysis:PMEasure<x>:SETup:I:
PROBe {C0 _ 001|C0 _ 002|C0 _ 005|C0 _ 01|
C0 _ 02|C0 _ 05|C0 _ 1|C0 _ 2|C0 _ 5|C1|
C2|C5|C10|C20|C50|C100|C200|C500|C1000|
C2000}
:ANALysis:PMEasure<x>:SETup:I:PROBe?
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:SETUP:I:
PROBE C0 _ 001
:ANALYSIS:PMEASURE1:SETUP:I:PROBE?
-> :ANALYSIS:PMEASURE1:SETUP:I:
PROBE C0 _ 001

:ANALysis:PMEasure<x>:SETup:RTRace

Function Sets or queries the deskewing source trace for power measurement.

Syntax :ANALysis:PMEasure<x>:SETup:
RTRace {I|U}
:ANALysis:PMEasure<x>:SETup:RTRace?
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:SETUP:RTRACE I
:ANALYSIS:PMEASURE1:SETUP:RTRACE?
-> :ANALYSIS:PMEASURE1:SETUP:
RTRACE I

:ANALysis:PMEasure<x>:SETup:U?

Function Queries all voltage input channel settings for voltage measurement.

Syntax :ANALysis:PMEasure<x>:SETup:U?
<x> = 1 to 2

:ANALysis:PMEasure<x>:SETup:U:DESKew

Function Sets or queries auto deskewing of the voltage input channel for power measurement.

Syntax :ANALysis:PMEasure<x>:SETup:U:
DESKew {<Time>}
:ANALysis:PMEasure<x>:SETup:U:
DESKew?
<x> = 1 to 2
<Time> = -1000.00 ns to 1000.00 ns (10 ps steps)

Example :ANALYSIS:PMEASURE1:SETUP:U:
DESKew 1NS
:ANALYSIS:PMEASURE1:SETUP:U:DESKew?
-> :ANALYSIS:PMEASURE1:SETUP:U:
DESKew 1.000E-09

:ANALysis:PMEasure<x>:SETup:U:PROBe

Function Sets or queries the probe attenuation of the voltage input channels for power measurement.

Syntax :ANALysis:PMEasure<x>:SETup:U:
PROBe {<NRf>}
:ANALysis:PMEasure<x>:SETup:U:PROBe?
<x> = 1 to 2
<NRf> = See the Features Guide for this information.

Example :ANALYSIS:PMEASURE1:SETUP:U:PROBE 1
:ANALYSIS:PMEASURE1:SETUP:U:PROBE?
-> :ANALYSIS:PMEASURE1:SETUP:U:
PROBE 1.000

:ANALysis:PMEasure<x>:TRANge

Function Sets or queries the measurement time period.

Syntax :ANALysis:PMEasure<x>:TRANge {<NRf>,
<NRf>}
:ANALysis:PMEasure<x>:TRANge?
<x> = 1 to 2
<NRf>, <NRf> = -5 to 5 (div), (10 div/display record length) steps

Example :ANALYSIS:PMEASURE1:TRANGE 5,-5
:ANALYSIS:PMEASURE1:TRANGE?
-> :ANALYSIS:PMEASURE1:
TRANGE 5.0000000E+00,-5.0000000E+00

:ANALysis:PMEasure<x>:UDPRoximal?

Function Queries all distal, mesial, and proximal settings.

Syntax :ANALysis:PMEasure<x>:UDPRoximal?
<x> = 1 to 2

5.3 ANALysis Group

:ANALysis:PMEasure<x>:UDProximal:MODE

Function Sets or queries the distal, mesial, and proximal point mode setting.

Syntax :ANALysis:PMEasure<x>:UDProximal:
MODE {PERCent|UNIT}
:ANALysis:PMEasure<x>:UDProximal:
MODE?
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:UDPROXIMAL:
MODE PERCENT
:ANALYSIS:PMEASURE1:UDPROXIMAL:MODE?
-> :ANALYSIS:PMEASURE1:UDPROXIMAL:
MODE PERCENT

:ANALysis:PMEasure<x>:UDProximal:PERCent

Function Sets or queries the distal, mesial, and proximal points as percentages.

Syntax :ANALysis:PMEasure<x>:UDProximal:
PERCent {<Nrf>,<Nrf>,<Nrf>}
:ANALysis:PMEasure<x>:UDProximal:
PERCent?
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:UDPROXIMAL:
PERCENT 10,50,90
:ANALYSIS:PMEASURE1:UDPROXIMAL:
PERCENT?
-> :ANALYSIS:PMEASURE1:UDPROXIMAL:
PERCENT 10,50,90

:ANALysis:PMEasure<x>:UDProximal:UNIT

Function Sets or queries the distal, mesial, and proximal points as voltages.

Syntax :ANALysis:PMEasure<x>:UDProximal:
UNIT {<Nrf>,<Nrf>,<Nrf>}
:ANALysis:PMEasure<x>:UDProximal:
UNIT?
<x> = 1 to 2

<Nrf>, <Nrf>, <Nrf> = See the Features Guide for this information.

Example :ANALYSIS:PMEASURE1:UDPROXIMAL:
UNIT -1,0,1
:ANALYSIS:PMEASURE1:UDPROXIMAL:UNIT?
-> :ANALYSIS:PMEASURE1:UDPROXIMAL:
UNIT -1.0000000E+00,0.0000000E+00,
1.0000000E+00

:ANALysis:PMEasure<x>:UMethod

Function Sets or queries the high and low points.

Syntax :ANALysis:PMEasure<x>:UMethod {AUTO|
MAXimum|HISTogram}
:ANALysis:PMEasure<x>:UMethod?
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:UMETHOD AUTO
:ANALYSIS:PMEASURE1:UMETHOD?
-> :ANALYSIS:PMEASURE1:UMETHOD AUTO

:ANALysis:PMEasure<x>:UNIT

Function Sets or queries the watt-hour unit.

Syntax :ANALysis:PMEasure<x>:UNIT {J|WH}
:ANALysis:PMEasure<x>:UNIT?
<x> = 1 to 2

Example :ANALYSIS:PMEASURE1:UNIT J
:ANALYSIS:PMEASURE1:UNIT?
-> :ANALYSIS:PMEASURE1:UNIT J

:ANALysis:PMEasure<x>:USER<y>?

Function Queries all automatic measurement settings for a Calc item.

Syntax :ANALysis:PMEasure<x>:USER<y>?
<x> = 1 to 2
<y> = 1 to 4

:ANALysis:PMEasure<x>:USER<y>:COUNT?

Function Queries the statistical processing count of the automatically measured value of a Calc item.

Syntax :ANALysis:PMEasure<x>:USER<y>:COUNT?
<x> = 1 to 2
<y> = 1 to 4

Example :ANALYSIS:PMEASURE:USER:COUNT?
-> :ANALYSIS:PMEASURE:USER:COUNT 1

:ANALysis:PMEasure<x>:USER<y>:DEFINE

Function Sets or queries the expression for the automatically measured value of a Calc item.

Syntax :ANALysis:PMEasure<x>:USER<y>:
DEFINE {<String>}
:ANALysis:PMEasure<x>:USER<y>:
DEFINE?
<x> = 1 to 2
<y> = 1 to 4
<String> = Up to 128 characters

Example :ANALYSIS:PMEASURE:USER:DEFINE "ABC"
:ANALYSIS:PMEASURE:USER:DEFINE?
-> :ANALYSIS:PMEASURE:USER:
DEFINE "ABC"

:ANALysis:PMEASURE<x>:USER<y>:{MAXimum|MEAN|MINimum|SDEViation}?

Function Queries a statistical value that is calculated on the automatically measured value of a Calc item.

Syntax :ANALysis:PMEASURE<x>:USER<y>:
{MAXimum|MEAN|MINimum|SDEViation}?
<x> = 1 to 2
<y> = 1 to 4

Example :ANALYSIS:PMEASURE:USER:MAXIMUM?
-> :ANALYSIS:PMEASURE:USER:
MAXIMUM 0.0

Description If the statistical value is immeasurable, the instrument returns "NAN" (not a number).

:ANALysis:PMEASURE<x>:USER<y>:NAME

Function Sets or queries the name of a Calc item.

Syntax :ANALysis:PMEASURE<x>:USER<y>:
NAME {<String>}
:ANALysis:PMEASURE<x>:USER<y>:NAME?
<x> = 1 to 2
<y> = 1 to 4
<String> = Up to 8 characters

Example :ANALYSIS:PMEASURE:USER:NAME "ABC"
:ANALYSIS:PMEASURE:USER:NAME?
-> :ANALYSIS:PMEASURE:USER:
NAME "ABC"

:ANALysis:PMEASURE<x>:USER<y>:STATE

Function Sets or queries the on/off status of automated measurement of a Calc item.

Syntax :ANALysis:PMEASURE<x>:USER<y>:
STATE {<Boolean>}
:ANALysis:PMEASURE<x>:USER<y>:STATE?
<x> = 1 to 2
<y> = 1 to 4

Example :ANALYSIS:PMEASURE:USER:STATE ON
:ANALYSIS:PMEASURE:USER:STATE?
-> :ANALYSIS:PMEASURE:USER:STATE 1

:ANALysis:PMEASURE<x>:USER<y>:UNIT

Function Sets or queries the unit of a Calc item.

Syntax :ANALysis:PMEASURE<x>:USER<y>:
UNIT {<String>}
:ANALysis:PMEASURE<x>:USER<y>:UNIT?
<x> = 1 to 2
<y> = 1 to 4
<String> = Up to 4 characters

Example :ANALYSIS:PMEASURE:USER:UNIT "ABC"
:ANALYSIS:PMEASURE:USER:UNIT?
-> :ANALYSIS:PMEASURE:USER:
UNIT "ABC"

:ANALysis:PMEASURE<x>:USER<y>:VALUE?

Function Queries the automatically measured value of a Calc item.

Syntax :ANALysis:PMEASURE<x>:USER<y>:
VALUE? {<Nrf>}
<x> = 1 to 2
<y> = 1 to 4
<Nrf> = 1 to 100000

Example :ANALYSIS:PMEASURE:USER:VALUE?
-> :ANALYSIS:PMEASURE:USER:VALUE 0.0

Description

- If the value is immeasurable, the instrument returns "NAN" (not a number).
- The <Nrf> is used to specify which iteration of automated measurement to query the measured value from.
- If <Nrf> is set to 1, the oldest measured value in the automated measurement memory is queried.
- If a measured value does not exist at the specified iteration, the instrument returns "NAN" (not a number).
- If <Nrf> is omitted, the most recent measured value is queried.

:ANALysis:WAIT?

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

Syntax :ANALysis:WAIT? {<Nrf>}
<Nrf> = 1 to 36000 (timeout value, 100 ms)

Example :ANALYSIS:WAIT? 10
-> :ANALYSIS:WAIT 1

Description If the execution of automated measurement is completed within the timeout period, 0 is returned. If it is not completed or if automated measurement is not being performed, 1 is returned. Even if you set a long timeout period, 0 is returned as soon as the automated measurement is completed.

5.4 ASETup Group

:ASETup:EXECute

Function Executes auto setup.

Syntax :ASETup:EXECute

Example :ASETUP:EXECUTE

:ASETup:UNDO

Function Undoes auto setup.

Syntax :ASETup:UNDO

Example :ASETUP:UNDO

5.5 CALibrate Group

:CALibrate?

Function Queries all calibration settings.

Syntax :CALibrate?

:CALibrate[:EXECute]

Function Executes calibration.

Syntax :CALibrate[:EXECute]

Example :CALIBRATE:EXECUTE

:CALibrate:MODE

Function Sets or queries the auto calibration on/off state.

Syntax :CALibrate:MODE {AUTO|OFF}

Example :CALIBRATE:MODE AUTO

:CALIBRATE:MODE?

-> :CALIBRATE:MODE AUTO

5.6 CHANnel Group

:CHANnel<x>?

Function Queries all vertical axis settings of a channel.

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

:CHANnel<x>:AScale[:EXECute]

Function Executes auto scaling of a channel.

Syntax CHANnel<x>:AScale[:EXECute]
<x> = 1 to 4

Example CHANNEL1:ASCALE:EXECUTE

Description Auto scaling is possible on each channel.

- V/div
Displayed so that the entire amplitude of the waveform can be seen without changing the vertical position.
- Offset
When the input coupling is set to AC
0 V
When the input coupling is set to DC
Center = (Max+Min)/2
- Trig Level
DC offset position

:CHANnel<x>:BWIDth

Function Sets or queries the input filter of the specified channel.

Syntax :CHANnel<x>:BWIDth {FULL|<Frequency>}
:CHANnel<x>:BWIDth?
<x> = 1 to 4
{<Frequency>} = See the Features Guide for this information.

Example :CHANNEL1:BWIDTH FULL
:CHANNEL1:BWIDTH?
-> :CHANNEL1:BWIDTH FULL

:CHANnel<x>:COUPling

Function Sets or queries the input coupling of the specified channel.

Syntax :CHANnel<x>:COUPling {AC|DC|DC50}
:CHANnel<x>:COUPling?
<x> = 1 to 4

Example :CHANNEL1:COUPLING AC
:CHANNEL1:COUPLING?
-> :CHANNEL1:COUPLING AC

:CHANnel<x>:DESKew

Function Sets or queries the deskewing of the specified channel.

Syntax :CHANnel<x>:DESKew {<Time>}
:CHANnel<x>:DESKew?
<x> = 1 to 4
<Time> = -1000.00 ns to 1000.00 ns (10 ps steps)

Example :CHANNEL1:DESKEW 1NS
:CHANNEL1:DESKEW?
-> :CHANNEL1:DESKEW 1.000E-09

:CHANnel<x>:DISPlay

Function Sets or queries the display on/off state of the specified channel.

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 Sets or queries the on/off state of the invert mode (waveform inversion display) .

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 specified channel.

Syntax :CHANnel<x>:LABel?
<x> = 1 to 4

:CHANnel<x>:LABel:DEFine

Function Sets or queries the waveform label of a channel.

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"

Description You can only use the characters and symbols on the keyboard that appears on the instrument's screen.

:CHANnel<x>:LABel:DISPlay

Function Sets or queries the display on/off state of the specified channel's waveform label.

Syntax :CHANnel<x>:LABel:
DISPlay {<Boolean>}
:CHANnel<x>:LABel:DISPlay?
<x> = 1 to 4

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

:CHANnel<x>:LSCale?

Function Queries all the linear scaling settings of the specified channel.

Syntax :CHANnel<x>:LSCale?
<x> = 1 to 4

:CHANnel<x>:LSCale:AVAlue

Function Sets or queries the scaling coefficient A.

Syntax :CHANnel<x>:LSCale:AVAlue {<NRf>}
:CHANnel<x>:LSCale:AVAlue?
<x> = 1 to 4
<NRf> = -1.0000E+31 to 1.0000E+31

Example :CHANNEL1:LSCALE:AVAlue 10
:CHANNEL1:LSCALE:AVAlue?
-> :CHANNEL1:LSCALE:
AVAlue 10.0000E+00

:CHANnel<x>:LSCale:BVALue

Function Sets or queries offset B.

Syntax :CHANnel<x>:LSCale:BVALue {<NRf>}
:CHANnel<x>:LSCale:BVALue?
<x> = 1 to 4
<NRf> = -1.0000E+31 to 1.0000E+31

Example :CHANNEL1:LSCALE:BVALue 10
:CHANNEL1:LSCALE:BVALue?
-> :CHANNEL1:LSCALE:
BVALue 10.0000E+00

:CHANnel<x>:LSCale:MODE

Function Sets or queries the on/off state of linear scaling.

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

Example :CHANNEL1:LSCALE:MODE ON
:CHANNEL1:LSCALE:MODE?
-> :CHANNEL1:LSCALE:MODE 1

:CHANnel<x>:LSCale:UNIT

Function Sets or queries a unit that is attached to linear scaling results.

Syntax :CHANnel<x>:LSCale:UNIT {<String>}
:CHANnel<x>:LSCale:UNIT?
<x> = 1 to 4
<String> = Up to 4 characters

Example :CHANNEL1:LSCALE:UNIT "EU"
:CHANNEL1:LSCALE:UNIT?
-> :CHANNEL1:LSCALE:UNIT "EU"

Description You can only use the characters and symbols on the keyboard that appears on the instrument's screen. The scaled values are never affected by the unit.

:CHANnel<x>:OFFSet

Function Sets or queries the offset voltage of the specified channel.

Syntax :CHANnel<x>:OFFSet {<Voltage>|
<Current>}
:CHANnel<x>:OFFSet?
<x> = 1 to 4
<Voltage>, <Current> = See the Features Guide for this information.

Example :CHANNEL1:OFFSET 0V
:CHANNEL1:OFFSET?
-> :CHANNEL1:OFFSET 0.000E+00

:CHANnel<x>:POSition

Function Sets or queries the vertical position of the specified channel.

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 1.00E+00

:CHANnel<x>:PROBe?

Function Queries all settings related to the probe attenuation of the specified channel.

Syntax :CHANnel<x>:PROBe?
<x> = 1 to 4

:CHANnel<x>:PROBe:DZCalibrate

Function Executes demagnetization and zero calibration of the specified channel.

Syntax :CHANnel<x>:PROBe:DZCalibrate
<x> = 1 to 4

Example :CHANNEL1:PROBE:DZCALIBRATE

5.6 CHANnel Group

:CHANnel<x>:PROBe[:MODE]

Function Sets or queries the probe attenuation of the specified channel.

Syntax :CHANnel<x>:PROBe[:MODE] {<NRF>|
C0 _ 001|C0 _ 002|C0 _ 005|C0 _ 01|
C0 _ 02|C0 _ 05|C0 _ 1|C0 _ 2|C0 _ 5|
C1|C2|C5|C10|C20|C50|C100|C200|C500|
C1000|C2000}
:CHANnel<x>:PROBe[:MODE]?
<x> = 1 to 4

Example :CHANNEL1:PROBE:MODE C0 _ 001
:CHANNEL1:PROBE:MODE?
-> :CHANNEL1:PROBE:MODE C0 _ 001

:CHANnel<x>:PROBe:PZCalibrate

Function Executes zero calibration of the specified channel's current probe.

Syntax :CHANnel<x>:PROBe:PZCalibrate
<x> = 1 to 4

Example :CHANNEL1:PROBE:PZCALIBRATE

:CHANnel<x>:VARIable

Function Sets or queries the specified channel's voltage scale in 0.01 V/div steps.

Syntax :CHANnel<x>:VARIable {<Voltage>|
<Current>}
:CHANnel<x>:VARIable?
<x> = 1 to 4
<Voltage>, <Current> = See the User's Manual
for this information.

Example :CHANNEL1:VARIABLE 1V
:CHANNEL1:VARIABLE?
-> :CHANNEL1:VARIABLE 1.000E+00

:CHANnel<x>:VDIV

Function Sets or queries the specified channel's voltage scale (V/div).

Syntax :CHANnel<x>:VDIV {<Voltage>|
<Current>}
:CHANnel<x>:VDIV?
<x> = 1 to 4
<Voltage>, <Current> = See the User's Manual
for this information.

Example :CHANNEL1:VDIV 2V
:CHANNEL1:VDIV?
-> :CHANNEL1:VDIV 2.000E+00

5.7 CLEar Group

:CLEar

Function Clears traces.

Syntax :CLEar

Example :CLEAR

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 communication settings.

Syntax :COMMunicate?

:COMMunicate:HEADer

Function Sets or queries whether a header is added to the response to a query (example with header: "CHANNEL1:PROBE:MODE 10"; example without header: "10").

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

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

:COMMunicate:LOCKout

Function Sets/clears local lockout.

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

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

:COMMunicate:OPSE (Operation Pending Status Enable register)

Function Sets or queries the overlap command that is used by the *OPC, *OPC?, and *WAI commands.

Syntax :COMMunicate:OPSE <Register>
:COMMunicate:OPSE?
<Register> = 0 to 65535 (see the figure for the :COMMunicate:WAIT? command)

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, so the response to the query only indicates 1 for bits 5, 6, 8, and 11.

:COMMunicate:OPSR? (Operation Pending Status Register)

Function Queries 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 or queries the commands that operate as overlap commands.

Syntax :COMMunicate:OVERlap <Register>
:COMMunicate:OVERlap?
<Register> = 0 to 65535

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, so the response to the query only indicates 1 for bits 5, 6, 8, and 11.
- For information about how to synchronize a program using :COMMunicate:OVERlap, see section 4.5.
- 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 the instrument to remote or local mode. On is remote mode.

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

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

:COMMunicate:VERBoSe

Function Sets or queries whether the response to a query is returned fully spelled out (e.g., CHANNEL1:PROBE:MODE 10) or in its abbreviated form (e.g., CHAN:PROB 10).

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

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

:COMMunicate:WAIT

Function Waits for a specified extended event to occur.

Syntax :COMMunicate:WAIT <Register>
<Register> = 0 to 65535 (Extended event register. For more information, see section 6.4.)

Example :COMMUNICATE:WAIT 65535

Description For information about how to synchronize a program using :COMMunicate:WAIT, see section 4.5.

:COMMunicate:WAIT?

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

Syntax :COMMunicate:WAIT? <Register>
<Register> = 0 to 65535 (Extended event register. For more information, see section 6.4.)

Example :COMMUNICATE:WAIT? 65535
-> 1

Description Operation pending status register and 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:

Media access not complete

When bit 8 (HST) = 1:

History search execution not complete

When bit 11 (SCH) = 1:

Edge/pattern search execution not complete

5.9 CURSor Group

:CURSor?

Function Queries all cursor measurement settings.
Syntax :CURSor?

:CURSor[:TY]?

Function Queries all cursor settings.
Syntax :CURSor[:TY]?

:CURSor[:TY]:DEGREE?

Function Queries all angle cursor settings.
Syntax :CURSor[:TY]:DEGREE?

:CURSor[:TY]:DEGREE:ALL

Function Collectively turns on or off the measured values of the angle cursor on the VT display.
Syntax :CURSor:TY:DEGREE:ALL {<Boolean>}
Example :CURSOR:TY:DEGREE:ALL ON

:CURSor[:TY]:DEGREE:D<x>?

Function Queries all angle cursor's angle settings.
Syntax :CURSor[:TY]:DEGREE:D<x>?
<x> = 1 to 2

:CURSor[:TY]:DEGREE:D<x>:STATE

Function Sets or queries the on/off state of the angle cursor's angle.
Syntax :CURSor[:TY]:DEGREE:D<x>:
STATE {<Boolean>}
:CURSor[:TY]:DEGREE:D<x>:STATE?
<x> = 1 to 2
Example :CURSOR:TY:DEGREE:D1:STATE ON
:CURSOR:TY:DEGREE:D1:STATE?
-> :CURSOR:TY:DEGREE:D1:STATE 1

:CURSor[:TY]:DEGREE:D<x>:VALUE?

Function Queries the angle cursor's angle.
Syntax :CURSor[:TY]:DEGREE:D<x>:VALUE?
<x> = 1 to 2
Example :CURSOR:TY:DEGREE:D1:VALUE?
-> :CURSOR:TY:DEGREE:D1:
VALUE -120.00000E+00

:CURSor[:TY]:DEGREE:DD?

Function Queries all settings related to the angle difference between the angle cursors.
Syntax :CURSor[:TY]:DEGREE:DD?

:CURSor[:TY]:DEGREE:DD:STATE

Function Sets or queries the on/off state of the angle difference ΔD value between the angle cursors.
Syntax :CURSor[:TY]:DEGREE:DD:
STATE {<Boolean>}
:CURSor[:TY]:DEGREE:DD:STATE?
Example :CURSOR:TY:DEGREE:DD:STATE ON
:CURSOR:TY:DEGREE:DD:STATE?
-> :CURSOR:TY:DEGREE:DD:STATE 1

:CURSor[:TY]:DEGREE:DD:VALUE?

Function Queries the angle difference ΔD value between the angle cursors.
Syntax :CURSor[:TY]:DEGREE:DD:VALUE?
Example :CURSOR:TY:DEGREE:DD:VALUE?
-> :CURSOR:TY:DEGREE:DD:
VALUE 180.00000E+00

:CURSor[:TY]:DEGREE:DV?

Function Queries all settings related to ΔV between the angle cursors.
Syntax :CURSor[:TY]:DEGREE:DV?

:CURSor[:TY]:DEGREE:DV:STATE

Function Sets or queries the on/off state of the ΔV value between the angle cursors.
Syntax :CURSor[:TY]:DEGREE:DV:
STATE {<Boolean>}
:CURSor[:TY]:DEGREE:DV:STATE?
Example :CURSOR:TY:DEGREE:DV:STATE ON
:CURSOR:TY:DEGREE:DV:STATE?
-> :CURSOR:TY:DEGREE:DV:STATE 1

:CURSor[:TY]:DEGREE:DV:VALue?

Function Queries the ΔV value between the angle cursors.

Syntax :CURSor[:TY]:DEGREE:DV:
VALue? {[V12]|V21}

Example :CURSOR:TY:DEGREE:DV:VALUE?
-> :CURSOR:TY:DEGREE:DV:
VALUE -6.2500000E-03
:CURSOR:TY:DEGREE:DV:VALUE? V21
-> :CURSOR:TY:DEGREE:DV:
VALUE 6.2500000E-03

Description • Both V12 and V21 options can be omitted.
If omitted, this command queries for the calculation result of $V1 - V2$.
• If V12 is selected, this command queries for the calculation result of $V1 - V2$.
• If V21 is selected, this command queries for the calculation result of $V2 - V1$.
• If linear scaling is set to ON, the scaled value will be queried.

:CURSor[:TY]:DEGREE:JUMP

Function Moves the angle cursor on the VT display on to the specified zoom waveform.

Syntax :CURSor[:TY]:DEGREE:JUMP {C1 _ Z1|
C1 _ Z2|C2 _ Z1|C2 _ Z2}

Example :CURSOR:TY:DEGREE:JUMP C1 _ Z1

Description This command moves the cursor to the center position of the zoom waveform. C1 and C2 are used to indicate Cursor1 and Cursor2.

:CURSor[:TY]:DEGREE:POSition<x>

Function Sets or queries the angle cursor position.

Syntax :CURSor[:TY]:DEGREE:
POSition<x> {<NRF>}
:CURSor[:TY]:DEGREE:POSition<x>?
<x> = 1 to 2
<NRF> = -5 to 5 (div), (10 div/display record length) steps

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

:CURSor[:TY]:DEGREE:REFeRence<x>

Function Sets or queries the start point (Reference1) or end point (Reference2) of the reference angle.

Syntax :CURSor[:TY]:DEGREE:
REFeRence<x> {<NRF>}
:CURSor[:TY]:DEGREE:REFeRence<x>?
<x> = 1 to 2
<NRF> = -5 to 5 (div), (10 div/display record length) steps

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

:CURSor[:TY]:DEGREE:RVALue

Function Sets or queries the reference angle.

Syntax :CURSor[:TY]:DEGREE:RVALue {<NRF>}
:CURSor[:TY]:DEGREE:RVALue?
<NRF> = 1 to 720

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

:CURSor[:TY]:DEGREE:TRACe

Function Sets or queries the source waveform of the angle cursor.

Syntax :CURSor[:TY]:DEGREE:TRACe {<NRF>|
ALL|MATH<x>}
:CURSor[:TY]:DEGREE:TRACe?
<NRF> = 1 to 4
<x> = 1 to 4

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

:CURSor[:TY]:DEGREE:UNIT

Function Sets or queries the angle cursor unit.

Syntax :CURSor[:TY]:DEGREE:UNIT {<String>}
:CURSor[:TY]:DEGREE:UNIT?
<String> = Up to 4 characters

Example :CURSOR:TY:DEGREE:UNIT "DEG"
:CURSOR:TY:DEGREE:UNIT?
-> :CURSOR:TY:DEGREE:UNIT "DEG"

:CURSor[:TY]:DEGREE:V<x>?

Function Queries all angle cursor's voltage settings.

Syntax :CURSor[:TY]:DEGREE:V<x>?
<x> = 1 to 2

5.9 CURSor Group

:CURSor[:TY]:DEGRee:V<x>:STATe

Function Sets or queries the on/off state of the angle cursor's voltage.

Syntax :CURSor[:TY]:DEGRee:V<x>:
STATe {<Boolean>}
:CURSor[:TY]:DEGRee:V<x>:STATe?
<x> = 1 to 2

Example :CURSOR:TY:DEGREE:V1:STATE ON
:CURSOR:TY:DEGREE:V1:STATE?
-> :CURSOR:TY:DEGREE:V1:STATE 1

:CURSor[:TY]:DEGRee:V<x>:VALue?

Function Queries the angle cursor's voltage.

Syntax :CURSor[:TY]:DEGRee:V<x>:VALue?
<x> = 1 to 2

Example :CURSOR:TY:DEGREE:V1:VALUE?
-> :CURSOR:TY:DEGREE:V1:
VALUE 10.000000E-03

:CURSor[:TY]:HORizontal?

Function Queries all ΔV cursor settings.

Syntax :CURSor[:TY]:HORizontal?

:CURSor[:TY]:HORizontal:ALL

Function Collectively turns on or off the measured values of the ΔV cursor on the VT display.

Syntax :CURSor:TY:HORizontal:
ALL {<Boolean>}

Example :CURSOR:TY:HORIZONTAL:ALL ON

:CURSor[:TY]:HORizontal:DV?

Function Queries all settings related to the vertical value between the ΔV cursors.

Syntax :CURSor[:TY]:HORizontal:DV?

:CURSor[:TY]:HORizontal:DV:STATe

Function Sets or queries the on/off state of the vertical value between the ΔV cursors.

Syntax :CURSor[:TY]:HORizontal:DV:
STATe {<Boolean>}
:CURSor[:TY]:HORizontal:DV:STATe?

Example :CURSOR:TY:HORIZONTAL:DV:STATE ON
:CURSOR:TY:HORIZONTAL:DV:STATE?
-> :CURSOR:TY:HORIZONTAL:DV:STATE 1

:CURSor[:TY]:HORizontal:DV:VALue?

Function Queries the vertical value between the ΔV cursors.

Syntax :CURSor[:TY]:HORizontal:DV:VALue?
Example :CURSOR:TY:HORIZONTAL:DV:VALUE?
-> :CURSOR:TY:HORIZONTAL:DV:
VALUE 3.0000000E+00

Description If linear scaling is set to ON, the scaled value will be queried.

:CURSor[:TY]:HORizontal:DVDT?

Function Queries all settings related to the ΔV cursor's $\Delta V/\Delta T$ value.

Syntax :CURSor[:TY]:HORizontal:DVDT?

:CURSor[:TY]:HORizontal:DVDT:STATe

Function Sets or queries the on/off state of the ΔV cursor's $\Delta V/\Delta T$ value.

Syntax :CURSor[:TY]:HORizontal:DVDT:
STATe {Boolean}
:CURSor[:TY]:HORizontal:DVDT:STATe?

Example :CURSOR:TY:HORIZONTAL:STATE ON
:CURSOR:TY:HORIZONTAL:STATE?
-> :CURSOR:TY:HORIZONTAL:STATE 1

:CURSor[:TY]:HORizontal:DVDT:VALue?

Function Queries the ΔV cursor's $\Delta V/\Delta T$ value.

Syntax :CURSor[:TY]:HORizontal:DVDT:VALue?
Example :CURSOR:TY:HORIZONTAL:VALUE?
-> :CURSOR:TY:HORIZONTAL:
VALUE 1.5E+03

Description If linear scaling is set to ON, the scaled value will be queried.

:CURSor[:TY]:HORizontal:POSition<x>

Function Sets or queries the ΔV cursor position.

Syntax :CURSor[:TY]:HORizontal:
POSition<x> {<NRf>}
:CURSor[:TY]:HORizontal:POSition<x>?
<x> = 1 to 2

<NRf> = -4 to 4 (div), (1/100 steps)
Example :CURSOR:TY:HORIZONTAL:POSITION1 -4
:CURSOR:TY:HORIZONTAL:POSITION1?
-> :CURSOR:TY:HORIZONTAL:
POSITION1 -4

Description <NRf> is significant to two decimal places.

:CURSor[:TY]:HORizontal:TRACe

Function Sets or queries the source waveform of the ΔV cursor.

Syntax :CURSor[:TY]:HORizontal:
TRACe {<NRf>|MATH<x>}
:CURSor[:TY]:HORizontal:TRACe?
<NRf> = 1 to 4
<x> = 1 to 4

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

:CURSor[:TY]:HORizontal:V<x>?

Function Queries all settings related to the ΔV cursor's vertical value.

Syntax :CURSor[:TY]:HORizontal:V<x>?
<x> = 1 to 2

:CURSor[:TY]:HORizontal:V<x>:STATe

Function Sets or queries the on/off state of the ΔV cursor's vertical value.

Syntax :CURSor[:TY]:HORizontal:V<x>:
STATe {<Boolean>}
:CURSor[:TY]:HORizontal:V<x>:STATe?
<x> = 1 to 2

Example :CURSOR:TY:HORIZONTAL:V1:STATE ON
:CURSOR:TY:HORIZONTAL:V1:STATE?
-> :CURSOR:TY:HORIZONTAL:V1:STATE 1

:CURSor[:TY]:HORizontal:V<x>:VALue?

Function Queries the ΔV cursor's vertical value.

Syntax :CURSor[:TY]:HORizontal:V<x>:VALue?
<x> = 1 to 2

Example :CURSOR:TY:HORIZONTAL:V1:VALUE?
-> :CURSOR:TY:HORIZONTAL:V1:
VALUE -1.5000000E+00

Description If linear scaling is set to ON, the scaled value will be queried.

:CURSor[:TY]:MARKer?

Function Queries all marker cursor settings.

Syntax :CURSor[:TY]:MARKer?

:CURSor[:TY]:MARKer:FORM

Function Sets or queries the display format of the marker cursor.

Syntax :CURSor[:TY]:MARKer:FORM {LINE|MARK}
:CURSor[:TY]:MARKer:FORM?

Example :CURSOR:TY:MARKER:FORM LINE
:CURSOR:TY:MARKER:FORM?
-> :CURSOR:TY:MARKER:FORM LINE

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

Function Queries all settings related to the specified marker cursor.

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

:CURSor[:TY]:MARKer:M<x>:ALL

Function Collectively turns on or off the measured values of the marker cursor on the VT display.

Syntax :CURSor:TY:MARKer:M<x>:
ALL {<Boolean>}
<x> = 1 to 4

Example :CURSOR:TY:MARKER:M1:ALL ON

:CURSor[:TY]:MARKer:M<x>:DT<y>?

Function Queries all settings related to the time value between the marker cursors.

Syntax :CURSor[:TY]:MARKer:M<x>:DT<y>?
<x> = 1 to 4
<y> = 1 to 4

:CURSor[:TY]:MARKer:M<x>:DT<y>:STATe

Function Sets or queries the on/off state of the time value between the marker cursors.

Syntax :CURSor[:TY]:MARKer:M<x>:DT<y>:
STATe {<Boolean>}
:CURSor[:TY]:MARKer:M<x>:DT<y>:
STATe?
<x> = 1 to 4
<y> = 1 to 4

Example :CURSOR:TY:MARKER:M:DT1:STATE ON
:CURSOR:TY:MARKER:M:DT1:STATE?
-> :CURSOR:TY:MARKER:M:DT1:STATE 1

:CURSor[:TY]:MARKer:M<x>:DT<y>:VALue?

Function Queries the time value between the marker cursors.

Syntax :CURSor[:TY]:MARKer:M<x>:DT<y>:
VALue?
<x> = 1 to 4
<y> = 1 to 4

Example :CURSOR:TY:MARKER:M:DT1:VALUE?
-> :CURSOR:TY:MARKER:M:DT1:
VALUE 0.0000000E+00

:CURSor[:TY]:MARKer:M<x>:DV<y>?

Function Queries all settings related to the vertical value between the marker cursors.

Syntax :CURSor[:TY]:MARKer:M<x>:DV<y>?
<x> = 1 to 4
<y> = 1 to 4

5.9 CURSOR Group

:CURSOR[:TY]:MARKER:M<x>:DV<y>:STATE

Function Sets or queries the on/off state of the vertical value between the marker cursors.

Syntax :CURSOR[:TY]:MARKER:M<x>:DV<y>:
STATE {<Boolean>}
:CURSOR[:TY]:MARKER:M<x>:DV<y>:
STATE?
<x> = 1 to 4
<y> = 1 to 4

Example :CURSOR:TY:MARKER:M:DV1:STATE ON
:CURSOR:TY:MARKER:M:DV1:STATE?
-> :CURSOR:TY:MARKER:M:DV1:STATE 1

:CURSOR[:TY]:MARKER:M<x>:DV<y>:VALUE?

Function Queries the vertical value between the marker cursors.

Syntax :CURSOR[:TY]:MARKER:M<x>:DV<y>:
VALUE?
<x> = 1 to 4
<y> = 1 to 4

Example :CURSOR:TY:MARKER:M:DV1:VALUE?
-> :CURSOR:TY:MARKER:M:DV1:
VALUE 500.00000E-03

Description If linear scaling is set to ON, the scaled value will be queried.

:CURSOR[:TY]:MARKER:M<x>:JUMP

Function Moves the marker cursor on the VT display on to the specified zoom waveform.

Syntax :CURSOR[:TY]:MARKER:M<x>:JUMP {Z1|
Z2}
<x> = 1 to 4

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

Description This command moves the cursor to the center position of the zoom waveform.

:CURSOR[:TY]:MARKER:M<x>:POSITION

Function Sets or queries the marker cursor's time value.

Syntax :CURSOR[:TY]:MARKER:M<x>:
POSITION {<NRF>}
:CURSOR[:TY]:MARKER:M<x>:POSITION?
<x> = 1 to 4
<NRF> = -5 to 5 (div), (10 div/display record
length) steps

Example :CURSOR:TY:MARKER:M1:POSITION -1
:CURSOR:TY:MARKER:M1:POSITION?
-> :CURSOR:TY:MARKER:M1:
POSITION -1.0000000

:CURSOR[:TY]:MARKER:M<x>:T?

Function Queries all settings related to the marker cursor's time value.

Syntax :CURSOR[:TY]:MARKER:M<x>:T?
<x> = 1 to 4

:CURSOR[:TY]:MARKER:M<x>:T:STATE

Function Sets or queries the on/off state of the marker cursor's time value.

Syntax :CURSOR[:TY]:MARKER:M<x>:T:
STATE {<Boolean>}
:CURSOR[:TY]:MARKER:M<x>:T:STATE?
<x> = 1 to 4

Example :CURSOR:TY:MARKER:M1:T:STATE ON
:CURSOR:TY:MARKER:M1:T:STATE?
-> :CURSOR:TY:MARKER:M1:T:STATE 1

:CURSOR[:TY]:MARKER:M<x>:T:VALUE?

Function Queries the marker cursor's time value.

Syntax :CURSOR[:TY]:MARKER:M<x>:T:VALUE?
<x> = 1 to 4
Example :CURSOR:TY:MARKER:M1:T:VALUE?
-> :CURSOR:TY:MARKER:M1:T:
VALUE -4.5000E-03

:CURSOR[:TY]:MARKER:M<x>:TRACE

Function Sets or queries the source waveform of the marker cursor.

Syntax :CURSOR[:TY]:MARKER:M<x>:
TRACE {<NRF>|MATH<y>|OFF}
:CURSOR[:TY]:MARKER:M<x>:TRACE?
<x> = 1 to 4
<NRF> = 1 to 4
<y> = 1 to 4

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

:CURSOR[:TY]:MARKER:M<x>:V?

Function Queries all settings related to the marker cursor's vertical value.

Syntax :CURSOR[:TY]:MARKER:M<x>:V?
<x> = 1 to 4

:CURSOR[:TY]:MARKER:M<x>:V:STATE

Function Sets or queries the on/off state of the marker cursor's vertical value.

Syntax :CURSOR[:TY]:MARKER:M<x>:V:
STATE {<Boolean>}
:CURSOR[:TY]:MARKER:M<x>:V:STATE?
<x> = 1 to 4

Example :CURSOR:TY:MARKER:M1:V:STATE ON
:CURSOR:TY:MARKER:M1:V:STATE?
-> :CURSOR:TY:MARKER:M1:V:STATE 1

:CURSor[:TY]:MARKer:M<x>:V:VALue?

Function Queries the marker cursor's vertical value.

Syntax :CURSor[:TY]:MARKer:M<x>:V:VALue?
<x> = 1 to 4

Example :CURSOR:TY:MARKer:M1:V:VALue?

-> :CURSOR:TY:MARKer:M1:V:

VALUE 1.5000E-03

Description If linear scaling is set to ON, the scaled value will be queried.

:CURSor[:TY]:TYPE

Function Sets or queries the cursor type.

Syntax :CURSor[:TY]:TYPE {OFF|HORizontal|
HAVertical|MARKer|VERTical|DEGRee}
:CURSor[:TY]:TYPE?

Example :CURSOR:TY:TYPE HORIZONTAL

:CURSOR:TY:TYPE?

-> :CURSOR:TY:TYPE HORIZONTAL

:CURSor[:TY]:VERTical?

Function Queries all ΔT cursor settings.

Syntax :CURSor[:TY]:VERTical?

:CURSor[:TY]:VERTical:ALL

Function Collectively turns on or off the measured values of the ΔT cursor on the VT display.

Syntax :CURSor:TY:VERTical:ALL {<Boolean>}

Example :CURSOR:TY:VERTICAL:ALL ON

:CURSor[:TY]:VERTical:DT?

Function Queries all settings related to the time value between the ΔT cursors.

Syntax :CURSor[:TY]:VERTical:DT?

:CURSor[:TY]:VERTical:DT:STATE

Function Sets or queries the on/off state of the time value between the ΔT cursors.

Syntax :CURSor[:TY]:VERTical:DT:
STATE {<Boolean>}
:CURSor[:TY]:VERTical:DT:STATE?

Example :CURSOR:TY:VERTICAL:DT:STATE ON

:CURSOR:TY:VERTICAL:DT:STATE?

-> :CURSOR:TY:VERTICAL:DT:STATE 1

:CURSor[:TY]:VERTical:DT:VALue?

Function Queries the time value between the ΔT cursors.

Syntax :CURSor[:TY]:VERTical:DT:VALue?

Example :CURSOR:TY:VERTICAL:DT:VALue?

-> :CURSOR:TY:VERTICAL:DT:

VALUE 2.50E-06

:CURSor[:TY]:VERTical:DV?

Function Queries all settings related to the vertical value between the ΔT cursors.

Syntax :CURSor[:TY]:VERTical:DV?

:CURSor[:TY]:VERTical:DV:STATE

Function Sets or queries the on/off state of the vertical value between the ΔT cursors.

Syntax :CURSor[:TY]:VERTical:DV:
STATE {<Boolean>}
:CURSor[:TY]:VERTical:DV:STATE?

Example :CURSOR:TY:VERTICAL:DV:STATE ON

:CURSOR:TY:VERTICAL:DV:STATE?

-> :CURSOR:TY:VERTICAL:DV:STATE 1

:CURSor[:TY]:VERTical:DV:VALue?

Function Queries the vertical value between the ΔT cursors.

Syntax :CURSor[:TY]:VERTical:DV:
VALue? {[V12]|V21}

Example :CURSOR:TY:VERTICAL:DV:VALue?

-> :CURSOR:TY:VERTICAL:DV:

VALUE -1.50E+03

:CURSOR:TY:VERTICAL:DV:VALue? V21

-> :CURSOR:TY:VERTICAL:DV:

VALUE 1.50E+03

Description • Both V12 and V21 options can be omitted.

If omitted, this command queries for the calculation result of $V1 - V2$.

• If V12 is selected, this command queries for the calculation result of $V1 - V2$.

• If V21 is selected, this command queries for the calculation result of $V2 - V1$.

• If linear scaling is set to ON, the scaled value will be queried.

:CURSor[:TY]:VERTical:DVDT?

Function Queries all settings related to the ΔT cursor's $\Delta V/\Delta T$ value.

Syntax :CURSor[:TY]:VERTical:DVDT?

:CURSor[:TY]:VERTical:DVDT:STATE

Function Sets or queries the on/off state of the ΔT cursor's $\Delta V/\Delta T$ value.

Syntax :CURSor[:TY]:VERTical:DVDT:
STATE {<Boolean>}

Example :CURSOR:TY:VERTICAL:DVDT:STATE ON

:CURSOR:TY:VERTICAL:DVDT:STATE?

-> :CURSOR:TY:VERTICAL:DVDT:STATE 1

5.9 CURSor Group

:CURSor[:TY]:VERTical:DVDT:VALue?

Function Queries the ΔT cursor's $\Delta V/\Delta T$ value.
Syntax :CURSor[:TY]:VERTical:DVDT:VALue?
Example :CURSOR:TY:VERTical:DVDT:VALue?
-> :CURSOR:TY:VERTical:DVDT:
VALUE 1.5E+03

Description If linear scaling is set to ON, the scaled value will be queried.

:CURSor[:TY]:VERTical:JUMP

Function Moves the vertical cursor on the VT display on to the specified zoom waveform.

Syntax :CURSor[:TY]:VERTical:JUMP {C1 _ Z1|
C1 _ Z2|C2 _ Z1|C2 _ Z2}

Example :CURSOR:TY:VERTICAL:JUMP C1 _ Z1

Description This command moves the cursor to the center position of the zoom waveform. C1 and C2 are used to indicate V cursors Cursor1 and Cursor2.

:CURSor[:TY]:VERTical:PERDt?

Function Queries all settings related to the $1/\Delta T$ value between the ΔT cursors.

Syntax :CURSor[:TY]:VERTical:PERDt?

:CURSor[:TY]:VERTical:PERDt:STATE

Function Sets or queries the on/off state of the $1/\Delta T$ value between the ΔT cursors.

Syntax :CURSor[:TY]:VERTical:PERDt:
STATE {<Boolean>}
:CURSor[:TY]:VERTical:PERDt:STATE?

Example :CURSOR:TY:VERTICAL:PERDT:STATE ON
:CURSOR:TY:VERTICAL:PERDT:STATE?
-> :CURSOR:TY:VERTICAL:PERDT:STATE 1

:CURSor[:TY]:VERTical:PERDt:VALue?

Function Queries the $1/\Delta T$ value between the ΔT cursors.

Syntax :CURSor[:TY]:VERTical:PERDt:VALue?
Example :CURSOR:TY:VERTICAL:PERDT:VALue?
-> :CURSOR:TY:VERTICAL:PERDT:
VALUE 2.50E+06

:CURSor[:TY]:VERTical:POSition<x>

Function Sets or queries the ΔT cursor position.

Syntax :CURSor[:TY]:VERTical:
POSition<x> {<NRf>}
:CURSor[:TY]:VERTical:POSition<x>?
<x> = 1 to 2
<NRf> = -5 to 5 (div), (10 div/display record
length) steps

Example :CURSOR:TY:VERTICAL:POSITION1 2
:CURSOR:TY:VERTICAL:POSITION1?
-> :CURSOR:TY:VERTICAL:
POSITION1 2.00E+00

:CURSor[:TY]:VERTical:T<x>?

Function Queries all settings related to the ΔT cursor's time value.

Syntax :CURSor[:TY]:VERTical:T<x>?
<x> = 1 to 2

:CURSor[:TY]:VERTical:T<x>:STATE

Function Sets or queries the on/off state of the ΔT cursor's time value.

Syntax :CURSor[:TY]:VERTical:T<x>:
STATE {<Boolean>}
:CURSor[:TY]:VERTical:T<x>:STATE?
<x> = 1 to 2

Example :CURSOR:TY:VERTICAL:T1:STATE ON
:CURSOR:TY:VERTICAL:T1:STATE?
-> :CURSOR:TY:VERTICAL:T1:STATE 1

:CURSor[:TY]:VERTical:T<x>:VALue?

Function Queries the ΔT cursor's time value.

Syntax :CURSor[:TY]:VERTical:T<x>:VALue?
<x> = 1 to 2
Example :CURSOR:TY:VERTICAL:T1:VALue?
-> :CURSOR:TY:VERTICAL:T1:
VALUE -2.50E-06

:CURSor[:TY]:VERTical:TRACe

Function Sets or queries the source waveform of the ΔT cursor.

Syntax :CURSor[:TY]:VERTical:TRACe {<NRf>|
ALL|MATH<x>}
:CURSor[:TY]:VERTical:TRACe?
<NRf> = 1 to 4
<x> = 1 to 4

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

:CURSor[:TY]:VERTical:V<x>?

Function Queries all settings related to the ΔT cursor's vertical value.

Syntax :CURSor[:TY]:VERTical:V<x>?
<x> = 1 to 2

:CURSor[:TY]:VERTical:V<x>:STATE

Function Sets or queries the on/off state of the ΔT cursor's vertical value.

Syntax :CURSor[:TY]:VERTical:V<x>:
STATE {<Boolean>}
:CURSor[:TY]:VERTical:V<x>:STATE?
<x> = 1 to 2

Example :CURSOR:TY:VERTICAL:V1:STATE ON
:CURSOR:TY:VERTICAL:V1:STATE?
-> :CURSOR:TY:VERTICAL:V1:STATE 1

:CURSor[:TY]:VERTical:V<x>:VALue?

Function Queries the ΔT cursor's vertical value.

Syntax :CURSor[:TY]:VERTical:V<x>:VALue?
<x> = 1 to 2

Example :CURSOR:TY:VERTICAL:V1:VALUE?

-> :CURSOR:TY:VERTICAL:V1:

VALUE 2.50E+03

5.10 DISPlay Group

:DISPlay?

Function Queries all display settings.

Syntax :DISPlay?

:DISPlay:ACCumulate?

Function Queries all accumulated waveform display settings.

Syntax :DISPlay:ACCumulate?

:DISPlay:ACCumulate:MODE

Function Sets or queries the accumulation mode.

Syntax :DISPlay:ACCumulate:MODE {COLOr|OFF|
PERSistence}
:DISPlay:ACCumulate:MODE?

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

:DISPlay:ACCumulate:PERSistence

Function Sets or queries the accumulation time.

Syntax :DISPlay:ACCumulate:
PERSistence {<Time>|INfinity}
:DISPlay:ACCumulate:PERSistence?
<Time> = 100 ms to 100 s

Example :DISPLAY:ACCUMULATE:
PERSISTENCE 100ms
:DISPLAY:ACCUMULATE:PERSISTENCE?
-> :DISPLAY:ACCUMULATE:
PERSISTENCE 100ms

:DISPlay:COLOr?

Function Queries all waveform color settings.

Syntax :DISPlay:COLOr?

:DISPlay:COLOr:{CHANnel<x>|MATH<x>}

Function Sets or queries a waveform color.

Syntax :DISPlay:COLOr:{CHANnel<x>|
MATH<x>} {BLUE|BGREEN|CYAN|DBLue|
GRAY|GREEN|LBLue|LGreen|MAGenta|
MGreen|ORANge|PINK|PURPle|RED|SPINK|
YELLow}
:DISPlay:COLOr:{CHANnel<x>|MATH<x>}?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

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

:DISPlay:COLOr:LState

Function Sets or queries the logic waveform state display color.

Syntax :DISPlay:COLOr:LState {BLUE|BGREEN|
CYAN|DBLue|GRAY|GREEN|LBLue|LGreen|
MAGenta|MGreen|ORANge|PINK|PURPle|
RED|SPINK|YELLow}
:DISPlay:COLOr:LState?

Example :DISPLAY:COLOR:LSTATE BLUE
:DISPLAY:COLOR:LSTATE?
-> :DISPLAY:COLOR:LSTATE BLUE

:DISPlay:COLOr:SERialbus<x>?

Function Queries the color settings of all trends of the specified serial bus.

Syntax :DISPlay:COLOr:SERialbus<x>?
<x> = 1 to 4

Description This command can be used on models with the serial bus option that supports trend display.

:DISPlay:COLOr:SERialbus<x>:TREND<y>

Function Sets or queries the serial bus trend color.

Syntax :DISPlay:COLOr:SERialbus<x>:
TREND<y> {BLUE|BGREEN|CYAN|DBLue|
GRAY|GREEN|LBLue|LGreen|MAGenta|
MGreen|ORANge|PINK|PURPle|RED|SPINK|
YELLow}
:DISPlay:COLOr:SERialbus<x>:
TREND<y>?
<x> = 1 to 4
<y> = 1 to 4

Example :DISPLAY:COLOR:SERIALBUS1:
TREND1 BLUE
:DISPLAY:COLOR:SERIALBUS1:TREND1?
-> :DISPLAY:COLOR:SERIALBUS1:
TREND1 BLUE

Description This command can be used on models with the serial bus option that supports trend display.

:DISPlay:FGRid

Function Sets or queries whether the fine grid is displayed.

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

Example :DISPLAY:FGRID ON
:DISPLAY:FGRID?
-> :DISPLAY:FGRID 1

:DISPlay:FORMat

Function Sets or queries the display format.

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

Example :DISPLAY:FORMAT AUTO
:DISPLAY:FORMAT?
-> :DISPLAY:FORMAT AUTO

:DISPlay:GRATicule

Function Sets or queries the graticule (grid).

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 display items.

Syntax :DISPlay:INTENsity?

:DISPlay:INTENsity[:WAVEform]

Function Sets or queries the waveform intensity.

Syntax :DISPlay:INTENsity[:WAVEform] {<NRf>}
:DISPlay:INTENsity[:WAVEform]?
<NRf> = 1 to 64

Example :DISPLAY:INTENSITY:WAVEFORM 10
:DISPLAY:INTENSITY:WAVEFORM?
-> :DISPLAY:INTENSITY:WAVEFORM 10

:DISPlay:INTENsity:{CURSor|GRID|MARKer|ZBOX}

Function Sets or queries the intensity of a display item.

Syntax :DISPlay:INTENsity:{CURSor|GRID|MARKer|ZBOX} {<NRf>}
:DISPlay:INTENsity:{CURSor|GRID|MARKer|ZBOX}?
<NRf> = 0 to 31

Example The following example is for the cursor.
:DISPLAY:INTENSITY:CUSOR 10
:DISPLAY:INTENSITY:CUSOR?
-> :DISPLAY:INTENSITY:CUSOR 10

:DISPlay:INTERpolate

Function Sets or queries the interpolation method.

Syntax :DISPlay:INTERpolate {OFF|LINE|PULSe|SINE}
:DISPlay:INTERpolate?

Example :DISPLAY:INTERPOLATE OFF
:DISPLAY:INTERPOLATE?
-> :DISPLAY:INTERPOLATE OFF

:DISPlay:MAPPING

Function Sets or queries the split screen waveform mapping mode.

Syntax :DISPlay:MAPPING {AUTO|MANual}
:DISPlay:MAPPING?

Example :DISPLAY:MAPPING AUTO
:DISPLAY:MAPPING?
-> :DISPLAY:MAPPING AUTO

:DISPlay:SMAPPING?

Function Queries all waveform mappings to the split screen.

Syntax :DISPlay:SMAPPING?

:DISPlay:SMAPPING:{CHANnel<x>|MATH<x>} (Set Mapping)

Function Sets or queries the specified waveform mapping to the split screen.

Syntax :DISPlay:SMAPPING:CHANnel<x> {<NRf>}
:DISPlay:SMAPPING:CHANnel<x>?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4
<NRf> = 1 to 8

Example :DISPLAY:SMAPPING:CHANNEL1 1
:DISPLAY:SMAPPING:CHANNEL1?
-> :DISPLAY:SMAPPING:CHANNEL1 1

:DISPlay:SVALue (Scale VALUE)

Function Sets or queries the on/off status of the scale value display.

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

Example :DISPLAY:SVALUE ON
:DISPLAY:SVALUE?
-> :DISPLAY:SVALUE 1

5.11 FFT Group

:FFT<x>?

Function Queries all FFT analysis settings.

Syntax :FFT<x>?
<x> = 1 to 2

:FFT<x>:AVERage?

Function Queries all settings related to the averaging of FFT analysis.

Syntax :FFT<x>:AVERage?
<x> = 1 to 2

:FFT<x>:AVERage:EWEight

Function Sets or queries the attenuation constant of exponential averaging of FFT analysis.

Syntax :FFT<x>:AVERage:EWEight {<NRf>}
:FFT<x>:AVERage:EWEight?
<x> = 1 to 2

<NRf> = 2 to 1024, 2n steps

Example :FFT1:AVERAGE:EWEIGHT 2
:FFT1:AVERAGE:EWEIGHT?
-> :FFT1:AVERAGE:EWEIGHT 2

:FFT<x>:DATA?

Function Queries all settings related to the FFT waveform data that will be sent.

Syntax :FFT<x>:DATA?
<x> = 1 to 2

:FFT<x>:DATA:BYTeorder

Function Sets or queries the transmission byte order when the FFT waveform data format is binary.

Syntax :FFT<x>:DATA:BYTeorder {LSBFirst|MSBFirst}
:FFT<x>:DATA:BYTeorder?
<x> = 1 to 2

Example :FFT1:DATA:BYTEORDER LSBFIRST
:FFT1:DATA:BYTEORDER?
-> :FFT1:DATA:BYTEORDER LSBFIRST

:FFT<x>:DATA:END

Function Sets or queries the end point of the FFT waveform data that will be sent.

Syntax :FFT<x>:DATA:END {<NRf>}
:FFT<x>:DATA:END?
<x> = 1 to 2
<NRf> = 0 to 1250000

Example :FFT1:DATA:END 125000
:FFT1:DATA:END?
-> :FFT1:DATA:END 125000

:FFT<x>:DATA:FORMat

Function Sets or queries the format of the FFT waveform data that will be sent.

Syntax :FFT<x>:DATA:FORMat {ASCIi|BINary}
:FFT<x>:DATA:FORMat?
<x> = 1 to 2

Example :FFT1:DATA:FORMAT ASCII
:FFT1:DATA:FORMAT?
-> :FFT:DATA:FORMAT ASCII

:FFT<x>:DATA:LENGth?

Function Queries the total number of data points of the FFT waveform that will be sent.

Syntax :FFT<x>:DATA:LENGth?
<x> = 1 to 2

Example :FFT1:DATA:LENGTH?
-> :FFT1:DATA:LENGTH 6251

:FFT<x>:DATA:SEND?

Function Queries the FFT waveform data.

Syntax :FFT<x>:DATA:SEND?
<x> = 1 to 2

Example :FFT1:DATA:SEND?
-> :FFT1:DATA:SEND #6
(6-digit number of bytes)(data sequence) or <NRf>,<NRf>,...

Description The output format of :FFT<x>:DATA:SEND? varies depending on the :FFT<x>:DATA:FORMat setting.

- When set to ASCIi, the information is returned in this form: <NRf>,<NRf>,...<NRf>
- When set to BINary, the information is returned in <block data> format. The data of each point is 4 bytes in length, and <block data> can be converted to computed values through floating-point conversion.

:FFT<x>:DATA:START

Function Sets or queries the start point of the FFT waveform data that will be sent.

Syntax :FFT<x>:DATA:START {<NRf>}
:FFT<x>:DATA:START?
<x> = 1 to 2
<NRf> = 0 to 125000

Example :FFT1:DATA:START 0
:FFT1:DATA:START?
-> :FFT1:DATA:START 0

:FFT<x>:DISPlay

Function Sets or queries whether FFT analysis is to be performed.

Syntax :FFT<x>:DISPlay {<Boolean>}
:FFT<x>:DISPlay?
<x> = 1 to 2

Example :FFT1:DISPLAY ON
:FFT1:DISPLAY?
-> :FFT1:DISPLAY 1

:FFT<x>:HORizontal?

Function Queries all horizontal axis settings of FFT analysis.

Syntax :FFT<x>:HORizontal?
<x> = 1 to 2

:FFT<x>:HORizontal:CSPan?

Function Queries all settings related to the center point and span of the horizontal axis in FFT analysis.

Syntax :FFT<x>:HORizontal:CSPan?
<x> = 1 to 2

:FFT<x>:HORizontal:CSPan:CENTer

Function Sets or queries the center value of the horizontal axis in FFT analysis.

Syntax :FFT<x>:HORizontal:CSPan:
CENTer {<Frequency>}
:FFT<x>:HORizontal:CSPan:CENTer?
<x> = 1 to 2
<Frequency> = 0 to 125 G(Hz)

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

:FFT<x>:HORizontal:CSPan:SPAN

Function Sets or queries the span of the horizontal axis in FFT analysis.

Syntax :FFT<x>:HORizontal:CSPan:
SPAN {<Frequency>}
:FFT<x>:HORizontal:CSPan:SPAN?
<x> = 1 to 2
<Frequency> = 0 to 125 G(Hz)

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

:FFT<x>:HORizontal:LRIGHT?

Function Queries all settings related to the left and right edges of the horizontal axis in FFT analysis.

Syntax :FFT<x>:HORizontal:LRIGHT?
<x> = 1 to 2

:FFT<x>:HORizontal:LRIGHT:RANGE

Function Sets or queries the range of the left and right edges of the horizontal axis in FFT analysis.

Syntax :FFT<x>:HORizontal:LRIGHT:
RANGE {<Frequency>,<Frequency>}
:FFT<x>:HORizontal:LRIGHT:RANGE?
<x> = 1 to 2
<Frequency> = 0 to 125 G(Hz)

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

:FFT<x>:HORizontal:MODE

Function Sets or queries the horizontal axis mode of FFT analysis.

Syntax :FFT<x>:HORizontal:MODE {AUTO|CSPan|
LRIGHT}
:FFT<x>:HORizontal:MODE?
<x> = 1 to 2

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

:FFT<x>:LENGth

Function Sets or queries the number of FFT points of FFT analysis.

Syntax :FFT<x>:LENGth {<NRf>}
:FFT<x>:LENGth?
<x> = 1 to 2
<NRf> = 1250, 2500, 12500, 25000, 125000,
250000, 1250000

Example :FFT1:LENGTH 1250
:FFT1:LENGTH?
-> :FFT1:LENGTH 1250

5.11 FFT Group

:FFT<x>:MEASure?

Function Queries all automated measurement settings of FFT analysis.

Syntax :FFT<x>:MEASure?
<x> = 1 to 2

:FFT<x>:MEASure:MARKer?

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

Syntax :FFT<x>:MEASure:MARKer?
<x> = 1 to 2

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

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

Syntax :FFT<x>:MEASure:MARKer:BASic?
<x> = 1 to 2

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

Function Collectively turns on or off all the basic marker cursor items of FFT analysis.

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

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

:FFT<x>:MEASure:MARKer[:BASic]:DFReque ncy?

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

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

:FFT<x>:MEASure:MARKer[:BASic]:DFReque ncy:STATE

Function Sets or queries the on/off state of the frequency between marker cursors of FFT analysis.

Syntax :FFT<x>:MEASure:MARKer[:BASic]:
DFRequency:STATE {<Boolean>}
:FFT<x>:MEASure:MARKer[:BASic]:
DFRequency:STATE?
<x> = 1 to 2

Example :FFT1:MEASURE:MARKER:BASIC:
DFREQUENCY:STATE ON
:FFT1:MEASURE:MARKER:BASIC:
DFREQUENCY:STATE?
-> :FFT1:MEASURE:MARKER:BASIC:
DFREQUENCY:STATE 1

:FFT<x>:MEASure:MARKer[:BASic]:DFReque ncy:VALue?

Function Queries the frequency between marker cursors of FFT analysis.

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

Example :FFT1:MEASURE:MARKER:BASIC:
DFREQUENCY:VALUE?
-> :FFT1:MEASURE:MARKER:BASIC:
DFREQUENCY:VALUE 1.000E+00

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

Function Queries all settings related to the level between marker cursors of FFT analysis.

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

:FFT<x>:MEASure:MARKer[:BASic]:DV:STA Te

Function Sets or queries the on/off state of the level between marker cursors of FFT analysis.

Syntax :FFT<x>:MEASure:MARKer[:BASic]:DV:
STATE {<Boolean>}
:FFT<x>:MEASure:MARKer[:BASic]:DV:
STATE?
<x> = 1 to 2

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

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

Function Queries the level between marker cursors of FFT analysis.

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

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

:FFT<x>:MEASure:MARKer[:BASic]:FREQuen cy<y>?

Function Queries all settings related to the specified marker cursor's frequency of FFT analysis.

Syntax :FFT<x>:MEASure:MARKer[:BASic]:
FREQuency<y>?
<x> = 1 to 2
<y> = 1 to 2

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

Function Sets or queries the on/off state of the specified marker cursor's frequency of FFT analysis.

Syntax :FFT<x>:MEASure:MARKer[:BASic]:FREQuency<y>:STATe {<Boolean>}
:FFT<x>:MEASure:MARKer[:BASic]:FREQuency<y>:STATe?
<x> = 1 to 2
<y> = 1 to 2

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

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

Function Queries the specified marker cursor's frequency of FFT analysis.

Syntax :FFT<x>:MEASure:MARKer[:BASic]:FREQuency<y>:VALue?
<x> = 1 to 2
<y> = 1 to 2

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

:FFT<x>:MEASure:MARKer[:BASic]:POSitio n<y>

Function Sets or queries the specified marker cursor's position of FFT analysis.

Syntax :FFT<x>:MEASure:MARKer[:BASic]:POSitio n<y> {<NRF>}
:FFT<x>:MEASure:MARKer[:BASic]:POSitio n<y>?
<x> = 1 to 2
<y> = 1 to 2
<NRF> = -5 to 5 (div)

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

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

Function Queries all settings related to the specified marker cursor's level of FFT analysis.

Syntax :FFT<x>:MEASure:MARKer[:BASic]:V<y>?
<x> = 1 to 2
<y> = 1 to 2

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

Function Sets or queries the on/off state of the specified marker cursor's level of FFT analysis.

Syntax :FFT<x>:MEASure:MARKer[:BASic]:V<y>:STATe {<Boolean>}
:FFT<x>:MEASure:MARKer[:BASic]:V<y>:STATe?
<x> = 1 to 2
<y> = 1 to 2

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

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

Function Queries the specified marker cursor's level of FFT analysis.

Syntax :FFT<x>:MEASure:MARKer[:BASic]:V<y>:VALue?
<x> = 1 to 2
<y> = 1 to 2

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

:FFT<x>:MEASure:MODE

Function Sets or queries the automated measurement mode of FFT analysis.

Syntax :FFT<x>:MEASure:MODE {MARKer|OFF|PARAmeter|PEAK}
:FFT<x>:MEASure:MODE?
<x> = 1 to 2

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

:FFT<x>:MEASure:PARAmeter?

Function Queries all settings related to the FFT analysis parameters.

Syntax :FFT<x>:MEASure:PARAmeter?
<x> = 1 to 2

:FFT<x>:MEASure:PARAmeter:OVERall?

Function Queries all settings related to the overall value, an FFT analysis parameter.

Syntax :FFT<x>:MEASure:PARAmeter:OVERall?
<x> = 1 to 2

5.11 FFT Group

:FFT<x>:MEASure:PARAmeter:OVERall:STAt

Function Sets or queries the on/off state of the overall value, an FFT analysis parameter.

Syntax :FFT<x>:MEASure:PARAmeter:OVERall:STATe {<Boolean>}
:FFT<x>:MEASure:PARAmeter:OVERall:STATe?
<x> = 1 to 2

Example :FFT1:MEASURE:PARAMETER:OVERALL:STATE ON
:FFT1:MEASURE:PARAMETER:OVERALL:STATE?
-> :FFT1:MEASURE:PARAMETER:OVERALL:STATE 1

:FFT<x>:MEASure:PARAmeter:OVERall:VALue?

Function Queries the overall value, an FFT analysis parameter.

Syntax :FFT<x>:MEASure:PARAmeter:OVERall:VALue?
<x> = 1 to 2

Example :FFT1:MEASURE:PARAMETER:OVERALL:VALUE?
-> :FFT1:MEASURE:PARAMETER:OVERALL:VALUE 2.27806E+00

:FFT<x>:MEASure:PEAK?

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

Syntax :FFT<x>:MEASure:PEAK?
<x> = 1 to 2

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

Function Queries all FFT analysis result settings.

Syntax :FFT<x>:MEASure:PEAK:DETail?
<x> = 1 to 2

:FFT<x>:MEASure:PEAK:DETail:LIST?

Function Queries all settings related to the FFT analysis result list.

Syntax :FFT<x>:MEASure:PEAK:DETail:LIST?
<x> = 1 to 2

:FFT<x>:MEASure:PEAK:DETail:LIST:ITEM?

Function Queries all items that will be displayed in the FFT analysis result list.

Syntax :FFT<x>:MEASure:PEAK:DETail:LIST:ITEM?
<x> = 1 to 2

Example :FFT1:MEASURE:PEAK:DETAIL:LIST:ITEM?
-> :FFT1:MEASURE:PEAK:DETAIL:LIST:ITEM? "No,Frequency,Peak"

:FFT<x>:MEASure:PEAK:DETail:LIST:MAXPeak:EXECute

Function Moves to the maximum peak value in the FFT analysis results.

Syntax :FFT<x>:MEASure:PEAK:DETail:LIST:MAXPeak:EXECute
<x> = 1 to 2

Example :FFT1:MEASURE:PEAK:DETAIL:LIST:MAXPeak:EXECUTE

:FFT<x>:MEASure:PEAK:DETail:LIST:NUMBER

Function Sets or queries the analysis number in the FFT analysis result list.

Syntax :FFT<x>:MEASure:PEAK:DETail:LIST:NUMBER {<NRf>}
:FFT<x>:MEASure:PEAK:DETail:LIST:NUMBER?
<x> = 1 to 2
<NRf> = 1 to 10

Example :FFT1:MEASURE:PEAK:DETAIL:LIST:NUMBER 1
:FFT1:MEASURE:PEAK:DETAIL:LIST:NUMBER?
-> :FFT1:MEASURE:PEAK:DETAIL:LIST:NUMBER 1

:FFT<x>:MEASure:PEAK:DETail:LIST:VALue?

Function Queries all the data for the specified analysis number in the FFT analysis result list.

Syntax :FFT<x>:MEASure:PEAK:DETail:LIST:VALue? {<NRf>}
<x> = 1 to 2
<NRf> = 1 to 10

Example :FFT1:MEASURE:PEAK:DETAIL:LIST:VALUE? 1
-> :FFT1:MEASURE:PEAK:DETAIL:LIST:VALUE "1, 0.000000 Hz,-10.00000 dBV"

:FFT<x>:MEASure:PEAK:EXCursion

Function Sets or queries the difference between peak and valley of the peak value of FFT analysis.

Syntax :FFT<x>:MEASure:PEAK:
EXCursion {<NRf>}
:FFT<x>:MEASure:PEAK:EXCursion?
<x> = 1 to 2
<NRf> = 0.0000000E+00 to 200.00000E+00

Example :FFT1:MEASURE:PEAK:EXCURSION 1
:FFT1:MEASURE:PEAK:EXCURSION?
-> :FFT1:MEASURE:PEAK:
EXCURSION 1.000E+00

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

Function Queries the peak frequency of FFT analysis.

Syntax :FFT<x>:MEASure:PEAK:FVALue?
<x> = 1 to 2

Example :FFT1:MEASURE:PEAK:FVALUE?
-> :FFT1:MEASURE:PEAK:
FVALUE 1.000E+00

:FFT<x>:MEASure:PEAK:RANGe

Function Sets or queries the measurement frequency range of the peak value of FFT analysis.

Syntax :FFT<x>:MEASure:PEAK:RANGe {<NRf>,
<NRf>}
:FFT<x>:MEASure:PEAK:RANGe?
<x> = 1 to 2

<NRf> = -5 to 5 (div), (10 div/display record length) steps

Example :FFT1:MEASURE:PEAK:RANGE 0,1
:FFT1:MEASURE:PEAK:RANGE?
-> :FFT1:MEASURE:PEAK:
RANGE 0.000E+00,1.000E+00

:FFT<x>:MEASure:PEAK:THReshold

Function Sets or queries the threshold level of the peak value of FFT analysis.

Syntax :FFT<x>:MEASure:PEAK:
THReshold {<NRf>}
:FFT<x>:MEASure:PEAK:THReshold?
<x> = 1 to 2

<NRf> = -1.0000E+31 to 1.0000E+31
Example :FFT1:MEASURE:PEAK:THRESHOLD 1
:FFT1:MEASURE:PEAK:THRESHOLD?
-> :FFT1:MEASURE:PEAK:
THRESHOLD 1.000E+00

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

Function Queries the peak value of FFT analysis.

Syntax :FFT<x>:MEASure:PEAK:VVALue?
<x> = 1 to 2

Example :FFT1:MEASURE:PEAK:VVALUE?
-> :FFT1:MEASURE:PEAK:
VVALUE 1.000E+00

:FFT<x>:MODE

Function Sets or queries the waveform display mode of FFT analysis.

Syntax :FFT<x>:MODE {AVERAge|MAXHold|
NORMAl}
:FFT<x>:MODE?
<x> = 1 to 2

Example :FFT1:MODE AVERAGE
:FFT1:MODE?
-> :FFT1:MODE AVERAGE

:FFT<x>:RANGe

Function Sets or queries the measurement source window of FFT analysis.

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

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

:FFT<x>:RPOSition

Function Sets or queries the center zoom point on the vertical axis in FFT analysis.

Syntax :FFT<x>:RPOSition {<NRf>}
:FFT<x>:RPOSition?
<x> = 1 to 2

<NRf> = -4 to 4 (div)

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

:FFT<x>:TRACe

Function Sets or queries the source waveform of FFT analysis.

Syntax :FFT<x>:TRACe {<NRf>|MATH<y>}
:FFT<x>:TRACe?
<x> = 1 to 2

<NRf> = 1 to 4
<y> of MATH<y> = 1 to 4

Example :FFT1:TRACE 1
:FFT1:TRACE?
-> :FFT1:TRACE 1

5.11 FFT Group

:FFT<x>:TYPE

Function Sets or queries the spectrum of FFT analysis.

Syntax :FFT<x>:TYPE {CH _MAG|CS _IMAG|
CS _LOGMAG|CS _MAG|CS _PHASE|
CS _REAL|LS _IMAG|LS _LOGMAG|
LS _MAG|LS _PHASE|LS _REAL|
PS _LOGMAG|PS _MAG|PSD _LOGMAG|
PSD _MAG|RS _LOGMAG|RS _MAG|
TF _IMAG|TF _LOGMAG|TF _MAG|
TF _PHASE|TF _REAL[,<Nrf>|MATH<y>]}

:FFT<x>:TYPE?

<x> = 1 to 2

<Nrf> = 1 to 4

<y> of MATH<y> = 1 to 4

Example :FFT1:TYPE CH _MAG,1

:FFT1:TYPE?

-> :FFT1:TYPE CH _MAG,1

Description • An error will occur if the user-defined computation (option) is not available.

- When the spectrum is set to CH_MAG|CS_IMAG|CS_LOGMAG|CS_MAG|CS_PHASE|CS_REAL|TF_IMAG|TF_LOGMAG|TF_MAG|TF_PHASE|TF_REAL, two source waveforms are required. Use :FFT<x>:TRACe to set the first waveform, and use <Nrf> or MATH<y> of this command to set the second waveform. If the spectrum is set to any other value, the source waveform specified with the :FFT<x>:TRACe command is valid.

:FFT<x>:UNIT?

Function Queries all FFT computation unit settings.

Syntax :FFT<x>:UNIT?

<x> = 1 to 2

:FFT<x>:UNIT:DBM?

Function Queries all settings for when the FFT computation unit is dBm.

Syntax :FFT<x>:UNIT:DBM?

<x> = 1 to 2

:FFT<x>:UNIT:DBM:IMPedance

Function Sets or queries the impedance when the FFT computation unit is dBm.

Syntax :FFT<x>:UNIT:DBM:IMPedance {<Nrf>}

<x> = 1 to 2

<Nrf> = 40 to 20000

Example :FFT1:UNIT:DBM:IMPEDANCE 50

:FFT1:UNIT:DBM:IMPEDANCE?

-> :FFT1:UNIT:DBM:IMPEDANCE 50

:FFT<x>:UNIT:MODE

Function Sets or queries whether an FFT computation unit will be attached automatically or manually.

Syntax :FFT<x>:UNIT:MODE {AUTO|DBM}

:FFT<x>:UNIT:MODE?

<x> = 1 to 2

Example :FFT1:UNIT:MODE AUTO

:FFT1:UNIT:MODE?

-> :FFT1:UNIT:MODE AUTO

:FFT<x>:VERTical?

Function Queries all vertical axis settings of FFT analysis.

Syntax :FFT<x>:VERTical?

<x> = 1 to 2

:FFT<x>:VERTical:LEVel

Function Sets or queries the vertical axis display position of FFT analysis.

Syntax :FFT<x>:VERTical:LEVel {<Nrf>}

:FFT<x>:VERTical:LEVel?

<x> = 1 to 2

<Nrf> = -1.0000E+31 to 1.0000E+31

Example :FFT1:VERTICAL:LEVEL 1

:FFT1:VERTICAL:LEVEL?

-> :FFT1:VERTICAL:LEVEL 1.000E+00

:FFT<x>:VERTical:MODE

Function Sets or queries the vertical axis mode of FFT analysis.

Syntax :FFT<x>:VERTical:MODE {AUTO|MANual}

:FFT<x>:VERTical:MODE?

<x> = 1 to 2

Example :FFT1:VERTICAL:MODE AUTO

:FFT1:VERTICAL:MODE?

-> :FFT1:VERTICAL:MODE AUTO

:FFT<x>:VERTical:SENSitivity

Function Sets or queries the vertical axis sensitivity of FFT analysis.

Syntax :FFT<x>:VERTical:SENSitivity {<Nrf>}

:FFT<x>:VERTical:SENSitivity?

<x> = 1 to 2

<Nrf> = 0 to 1.0000E+31

Example :FFT1:VERTICAL:SENSITIVITY 1

:FFT1:VERTICAL:SENSITIVITY?

-> :FFT1:VERTICAL:
SENSITIVITY 1.000E+00

:FFT<x>:VTDisplay

Function Sets or queries the on/off state of the VT waveform display of FFT analysis.

Syntax :FFT<x>:VTDisplay {<Boolean>}
:FFT<x>:VTDisplay?
<x> = 1 to 2

Example :FFT1:VTDISPLAY ON
:FFT1:VTDISPLAY?
-> :FFT1:VTDISPLAY 1

:FFT<x>:WINDow

Function Sets or queries the window function of FFT analysis.

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

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

5.12 FILE Group

:FILE?

Function Queries all data storage settings.
Syntax :FILE?

:FILE:COPY:ABORT

Function Aborts file copying.
Syntax :FILE:COPY:ABORT
Example :FILE:COPY:ABORT

:FILE:COPY:CDIRECTORY

Function Changes the file copy destination directory.
Syntax :FILE:COPY:CDIRECTORY {<String>}
<String> = See the Features Guide for this information.
Example :FILE:COPY:CDIRECTORY "UTIL"
(Relative path designation)

:FILE:COPY:DRIVE

Function Sets the storage device to copy files to.
Syntax :FILE:COPY:DRIVE {FLASHmem|NETWork|
USB,<Nrf>}
<Nrf> = 0 to 1
Example :FILE:COPY:DRIVE FLASHMEM
Description The <Nrf> for USB can be omitted if the drive is
not partitioned or divided by LUN.

:FILE:COPY[:EXECute]

Function Executes file copying. This is an overlap
command.
Syntax :FILE:COPY[:EXECute] {<String>}
<String> = See the Features Guide for this
information.
Example :FILE:COPY:EXECUTE "DATA.PNG"

:FILE:COPY:PATH?

Function Queries the file copy destination directory.
Syntax :FILE:COPY:PATH?
Example :FILE:COPY:PATH?
-> :FILE:COPY:
PATH "PATH = FLASHMEM/UTIL"

:FILE:DELeTe:{AHISTogram|ASCIi|BINary| BMP|FFT|HLISt|JPEG|MEASure|PNG|SBUS|SET up|SNAP|ZPOLygon|ZWAVE}[:EXECute]

Function Deletes a specific type of file. This is an overlap
command.
Syntax :FILE:DELeTe:{AHISTogram|ASCIi|
BINary|BMP|FFT|HLISt|JPEG|MEASure|
PNG|SBUS|SETup|SNAP|ZPOLygon|
ZWAVE}[:EXECute] {<Filename>}

Example The following example is for waveform data.
:FILE:DELeTe:AHISTOGRAM:
EXECUTE "DATA"

Description Use the :FILE:DIReCTory:DRIVE to select the
target storage device.

:FILE[:DIReCTory]:CDIRECTORY

Function Changes the current directory.
Syntax :FILE[:DIReCTory]:
CDIRECTORY {<String>}
<String> = See the Features Guide for this
information.
Example :FILE:DIReCTory:CDIRECTORY "UTIL"

:FILE[:DIReCTory]:DRIVE

Function Sets the current storage device.
Syntax :FILE[:DIReCTory]:DRIVE {FLASHmem|
NETWork|USB,<Nrf>}
<Nrf> = 0 to 1
Example :FILE:DIReCTory:DRIVE FLASHMEM
Description The <Nrf> for USB can be omitted if the drive is
not partitioned or divided by LUN.

:FILE[:DIReCTory]:FREE?

Function Queries the free space on the current storage
device that is being operated on in bytes.
Syntax :FILE[:DIReCTory]:FREE?
Example :FILE:DIReCTory:FREE?
-> :FILE:DIReCTory:FREE 65536

:FILE[:DIReCTory]:MDIRECTORY

Function Creates a directory in the current location. This is
an overlap command.
Syntax :FILE[:DIReCTory]:MDIRECTORY
{<String>}
<String> = See the Features Guide for this
information.
Example :FILE:DIReCTory:MDIRECTORY "DIR2"

:FILE[:DIRectory]:PATH?

Function Queries the current directory.
 Syntax :FILE[:DIRectory]:PATH?
 Example :FILE:DIReCTORY:PATH?
 -> :FILE:DIReCTORY:
 PATH "PATH = FLASHMEM/UTIL"

:FILE:LOAD:BINArY:ABORt

Function Aborts the loading of waveform data.
 Syntax :FILE:LOAD:BINArY:ABORt
 Example :FILE:LOAD:BINArY:ABORt

:FILE:LOAD:BINArY[:EXECute]

Function Executes the loading of waveform data. This is an overlap command.
 Syntax :FILE:LOAD:BINArY[:EXECute] {<Filename>{,ACQMemory|REFERENCE<x>}}
 <x> = 1 to 2
 Example :FILE:LOAD:BINArY:EXECUTE "DATA"

:FILE:LOAD:{SETUp|SNAP|ZPOLygon<x>|ZWAVE<x>}:ABORt

Function Aborts the loading of a specific type of file.
 Syntax :FILE:LOAD:{SETUp|SNAP|ZPOLygon<x>|ZWAVE<x>}:ABORt
 <x> = 1 to 4
 Example The following example is for setup data.
 :FILE:LOAD:SETUp:ABORt

:FILE:LOAD:{SETUp|SNAP|ZPOLygon<x>|ZWAVE<x>}:EXECute]

Function Executes the loading of a specific type of file. This is an overlap command.
 Syntax :FILE:LOAD:{SETUp|SNAP|ZPOLygon<x>|ZWAVE<x>}:EXECute] {<Filename>}
 <x> = 1 to 4
 Example The following example is for setup data.
 :FILE:LOAD:SETUp:EXECUTE "DATA"

:FILE:MOVE:ABORt

Function Aborts file moving.
 Syntax :FILE:MOVE:ABORt
 Example :FILE:MOVE:ABORt

:FILE:MOVE:CDIRectory

Function Changes the file move destination directory.
 Syntax :FILE:MOVE:CDIRectory {<String>}
 <String> = See the Features Guide for this information.
 Example :FILE:MOVE:CDIRectory "UTIL"

:FILE:MOVE:DRIVE

Function Sets the storage device to move files to.
 Syntax :FILE:MOVE:DRIVE {FLASHmem|USB,<Nrf>}
 <Nrf> = 0 to 1
 Example :FILE:MOVE:DRIVE FLASHMEM
 Description The <Nrf> for USB can be omitted if the drive is not partitioned or divided by LUN.

:FILE:MOVE[:EXECute]

Function Executes file moving. This is an overlap command.
 Syntax :FILE:MOVE[:EXECute] {<String>}
 <String> = See the Features Guide for this information.
 Example :FILE:MOVE:EXECUTE "DATA.PNG"

:FILE:MOVE:PATH?

Function Queries the file move destination directory.
 Syntax :FILE:MOVE:PATH?
 Example :FILE:MOVE:PATH?
 -> :FILE:MOVE:
 PATH "PATH = FLASHMEM/UTIL"

:FILE:PROTeCt[:EXECute]

Function Turns on or off the file protection.
 Syntax :FILE:PROTeCt[:EXECute] {<String>,<Boolean>}
 <String> = See the Features Guide for this information.
 Example :FILE:PROTeCt:EXECUTE "DATA.PNG",ON

:FILE:REName[:EXECute]

Function Renames the file.
 Syntax :FILE:REName[:EXECute] {<String>,<String>}
 <String> = See the Features Guide for this information.
 Example :FILE:REName:EXECUTE "DATA.PNG",
 "000.PNG"

:FILE:SAVE?

Function Queries all file save settings.
 Syntax :FILE:SAVE?

:FILE:SAVE:{AHISTogram|ASCIi|BINArY|FFT|HARMonics|HLISt|SBUS|ZWAVE}?

Function Queries all the settings related to the saving of a specific type of file.
 Syntax :FILE:SAVE:{AHISTogram|ASCIi|BINArY|FFT|HARMonics|HLISt|MEASure|SBUS|SETUp|SNAP|ZWAVE}?

5.12 FILE Group

:FILE:SAVE:{AHISTogram|ASCIi|BINary|FFT|HARMonics|HLISt|MEASure|SBUS|SETup|SNAP|ZWAVE}:ABORT

Function Aborts the saving of a specific type of file.

Syntax :FILE:SAVE:{AHISTogram|ASCIi|BINary|FFT|HARMonics|HLISt|MEASure|SBUS|SETup|SNAP|ZWAVE}:ABORT

Example The following example is for waveform data.
:FILE:SAVE:BINARY:ABORT

:FILE:SAVE:{AHISTogram|ASCIi|BINary|FFT|HARMonics|HLISt|MEASure|SBUS|SETup|SNAP|ZWAVE}:EXECute

Function Executes the saving of a specific type of file. This is an overlap command.

Syntax :FILE:SAVE:{AHISTogram|ASCIi|BINary|FFT|HARMonics|HLISt|MEASure|SBUS|SETup|SNAP|ZWAVE}:EXECute

Example The following example is for waveform data.
:FILE:SAVE:BINARY:EXECUTE "DATA"

:FILE:SAVE:{AHISTogram|FFT|HARMonics|SBUS|ZWAVE}:SELEct

Function Sets or queries the area for saving a specific type of file.

Syntax :FILE:SAVE:AHISTogram:SELEct {<NRf>}
:FILE:SAVE:AHISTogram:SELEct?
<NRf> = 1 to 2 (1 to 4 for SBUS and ZWAVE)

Example The following example is for FFT.
:FILE:SAVE:FFT:SELECT 1
:FILE:SAVE:FFT:SELECT?
-> :FILE:SAVE:FFT:SELECT 1

:FILE:SAVE:ANAMing

Function Sets or queries the on/off state of the auto generation of the name of the file to be saved.

Syntax :FILE:SAVE:ANAMing {DATE|DATE2|NUMBERing|OFF|ON}
:FILE:SAVE:ANAMing?

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

:FILE:SAVE:{ASCIi|BINary}:COMPression

Function Sets or queries the save compression method for a specific type of file.

Syntax :FILE:SAVE:{ASCIi|BINary}:COMPression {DECimation|OFF|PTOPeak}
:FILE:SAVE:{ASCIi|BINary}:COMPression?

Example The following example is for waveform data.
:FILE:SAVE:BINARY:
COMPRESSION DECIMATION
:FILE:SAVE:BINARY:COMPRESSION?
-> :FILE:SAVE:BINARY:
COMPRESSION DECIMATION

:FILE:SAVE:{ASCIi|BINary|SBUS}:HISTory

Function Sets or queries the save method for the specified history waveform.

Syntax :FILE:SAVE:{ASCIi|BINary|SBUS}:HISTory {ALL|ONE}
:FILE:SAVE:{ASCIi|BINary|SBUS}:HISTory?

Example The following example is for waveform data.
:FILE:SAVE:BINARY:HISTORY ALL
:FILE:SAVE:BINARY:HISTORY?
-> :FILE:SAVE:BINARY:HISTORY ALL

:FILE:SAVE:{ASCIi|BINary}:LENGth

Function Sets or queries the number of data points to save when compressing or sampling (decimating) various types of data.

Syntax :FILE:SAVE:{ASCIi|BINary}:LENGth {<NRf>}
:FILE:SAVE:{ASCIi|BINary}:LENGth?
<NRf> = See the Features Guide for this information.

Example The following example is for waveform data.
:FILE:SAVE:BINARY:LENGTH 1250
:FILE:SAVE:BINARY:LENGTH?
-> :FILE:SAVE:BINARY:LENGTH 1250

:FILE:SAVE:{ASCIi|BINary}:RANGE

Function Sets or queries the window for saving a specific type of data.

Syntax :FILE:SAVE:{ASCIi|BINary}:RANGE {MAIN|Z1|Z2}
:FILE:SAVE:{ASCIi|BINary}:RANGE?

Example The following example is for waveform data.
:FILE:SAVE:BINARY:RANGE MAIN
:FILE:SAVE:BINARY:RANGE?
-> :FILE:SAVE:BINARY:RANGE MAIN

:FILE:SAVE:AScii:TINformation (Time Information)

Function Sets or queries whether waveform data is saved with time information (ON) or without it (OFF).

Syntax :FILE:SAVE:AScii:
TINformation {<Boolean>}
:FILE:SAVE:AScii:TINformation?

Example :FILE:SAVE:AScii:TINFORMATION ON
:FILE:SAVE:AScii:TINFORMATION?
-> :FILE:SAVE:AScii:TINFORMATION 1

:FILE:SAVE:{AScii|BINary}:TRACe

Function Sets or queries the waveform for saving a specific type of data.

Syntax :FILE:SAVE:{AScii|BINary}:
TRACe {<NRf>|ALL|MATH<x>}
:FILE:SAVE:{AScii|BINary}:TRACe?
<NRf> = 1 to 4
<x> = 1 to 4

Example The following example is for waveform data.
:FILE:SAVE:BINARY:TRACE 1
:FILE:SAVE:BINARY:TRACE?
-> :FILE:SAVE:AScii:TRACE 1

:FILE:SAVE:COMMeNt

Function Sets or queries the comment for the data to be saved.

Syntax :FILE:SAVE:COMMeNt {<String>}
:FILE:SAVE:COMMeNt?
<String> = Up to 128 characters

Example :FILE:SAVE:COMMENT "THIS IS TEST."
:FILE:SAVE:COMMENT?
-> :FILE:SAVE:
COMMENT "THIS IS TEST."

Description You can only use the characters and symbols on the keyboard that appears on the instrument's screen.

:FILE:SAVE:FFT:FINformation (FrequencyInformation)

Function Sets or queries whether FFT data is saved with frequency information (ON) or without it (OFF).

Syntax :FILE:SAVE:FFT:
FINformation {<Boolean>}
:FILE:SAVE:FFT:FINformation?

Example :FILE:SAVE:FFT:FINFORMATION ON
:FILE:SAVE:FFT:FINFORMATION?
-> :FILE:SAVE:FFT:FINFORMATION 1

:FILE:SAVE:NAME

Function Sets or queries the file name for the data to be saved.

Syntax :FILE:SAVE:NAME {<Filename>}
:FILE:SAVE:NAME?

Example :FILE:SAVE:NAME "CASE1"
:FILE:SAVE:NAME?
-> :FILE:SAVE:NAME "CASE1"

5.13 GONogo Group

:GONogo?

Function Queries all GO/NO-GO determination settings.
Syntax :GONogo?

:GONogo:ABORT

Function Aborts GO/NO-GO determination.
Syntax :GONogo:ABORT
Example :GONOGO:ABORT

:GONogo:ACTion?

Function Queries all settings for the action that is performed when the judgment is NO-GO and the reference for the judgment.
Syntax :GONogo:ACTion?

:GONogo:ACTion:BUZZer

Function Sets or queries whether a warning sound is to be generated when the judgment is NO-GO.
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 or queries whether a hard copy is output to the built-in printer (option) or an external printer when the judgment is NO-GO.
Syntax :GONogo:ACTion:HCOPY {<Boolean>}
:GONogo:ACTion:HCOPY?
Example :GONOGO:ACTION:HCOPY ON
:GONOGO:ACTION:HCOPY?
-> :GONOGO:ACTION:HCOPY 1

:GONogo:ACTion:MAIL?

Function Queries all settings related to the mail notification that is made when the judgment is NO-GO.
Syntax :GONogo:ACTion:MAIL?

:GONogo:ACTion:MAIL:COUNT

Function Sets or queries the upper limit to the number of mail notifications made when the judgment is NO-GO.
Syntax :GONogo:ACTion:MAIL:COUNT {<NRF>}
:GONogo:ACTion:MAIL:COUNT?
<NRF> = 1 to 1000
Example :GONOGO:ACTION:MAIL:COUNT 100
:GONOGO:ACTION:MAIL:COUNT?
-> :GONOGO:ACTION:MAIL:COUNT 100

:GONogo:ACTion:MAIL:MODE

Function Sets or queries whether a mail notification is to be made when the judgment is NO-GO.
Syntax :GONogo:ACTion:MAIL:MODE {<Boolean>}
:GONogo:ACTion:MAIL:MODE?
Example :GONOGO:ACTION:MAIL:MODE ON
:GONOGO:ACTION:MAIL:MODE?
-> :GONOGO:ACTION:MAIL:MODE 1

:GONogo:ACTion:SAVE

Function Sets or queries whether the event is to be saved to the storage device when the judgment is NO-GO.
Syntax :GONogo:ACTion:SAVE {<Boolean>}
:GONogo:ACTion:SAVE?
Example :GONOGO:ACTION:SAVE ON
:GONOGO:ACTION:SAVE?
-> :GONOGO:ACTION:SAVE 1
Description Use :FILE:DIRECTORY:DRIVE to set or query the type of storage device.

:GONogo:COUNT?

Function Queries the number of GO/NO-GO determinations that were performed.
Syntax :GONogo:COUNT?
Example :GONOGO:COUNT?
-> :GONOGO:COUNT 100

:GONogo:EXECute

Function Executes GO/NO-GO determination. This is an overlap command.
Syntax :GONogo:EXECute
Example :GONOGO:EXECUTE

:GONogo:LOGic

Function Sets or queries the GO/NO-GO determination type.
Syntax :GONogo:LOGic {AND|OFF|OR}
:GONogo:LOGic?
Example :GONOGO:LOGIC AND
:GONOGO:LOGIC?
-> :GONOGO:LOGIC AND

:GONogo:NGCount?

Function Queries the GO/NO-GO determination NO-GO count.
Syntax :GONogo:NGCount?
Example :GONOGO:NGCOUNT?
-> :GONOGO:NGCOUNT 5

:GONogo:NGStopcount

Function Sets or queries the number of NO-GO results.

Syntax :GONogo:NGStopcount {<NRf>|INFinity}
:GONogo:NGStopcount?
<NRf> = 1 to 1000

Example :GONOGO:NGSTOPCOUNT 100
:GONOGO:NGSTOPCOUNT?
-> :GONOGO:NGSTOPCOUNT 100

:GONogo:STOPcount

Function Sets or queries the number of determination results acquired.

Syntax :GONogo:STOPcount {<NRf>|INFinity}
:GONogo:STOPcount?
<NRf> = 1 to 1000000

Example :GONOGO:STOPCOUNT 1000
:GONOGO:STOPCOUNT?
-> :GONOGO:STOPCOUNT 1000

:GONogo:WAIT?

Function Waits for the completion of GO/NO-GO determination with a timeout.

Syntax :GONogo:WAIT? {<NRf>}
<NRf> = 0 to 864000 (10 ms steps)

Example (Set the timeout period to 5 s.)
:GONOGO:WAIT? 500
-> :GONOGO:WAIT 0

Description The instrument returns 0 if the operation finishes before the timer expires and returns 1 if the timer expires.

:GONogo[:ZPARAMeter]:?

Function Queries all zone or parameter determination settings.

Syntax :GONogo[:ZPARAMeter]:?

:GONogo[:ZPARAMeter]:NUMBER<x>?

Function Queries all settings related to the zone or parameter determination conditions.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>?
<x> = 1 to 4

:GONogo[:ZPARAMeter]:NUMBER<x>:CAUSE?

Function Queries whether the specified waveform parameter for zone or parameter determination is the cause of a NO-GO judgment.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:
CAUSE?
<x> = 1 to 4

Example :GONOGO:ZPARAMETER:NUMBER1:CAUSE?
-> :GONOGO:ZPARAMETER:NUMBER1:
CAUSE 1

Description When the parameter is the cause of a NO-GO result, the instrument returns 1. Otherwise, the instrument returns 0.

:GONogo[:ZPARAMeter]:NUMBER<x>:CONDiti on

Function Sets or queries the specified waveform parameter's reference condition for zone or parameter determination.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:
CONDition {IN|OUT}
:GONogo[:ZPARAMeter]:NUMBER<x>:
CONDition?
<x> = 1 to 4

Example :GONOGO:ZPARAMETER:NUMBER1:
CONDITION IN
:GONOGO:ZPARAMETER:NUMBER1:
CONDITION?
-> :GONOGO:ZPARAMETER:NUMBER1:
CONDITION IN

:GONogo[:ZPARAMeter]:NUMBER<x>:MODE

Function Sets or queries the mode of the specified condition.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:
MODE {OFF|PARAMeter|POLYgon|
RECTangle|WAVE}
:GONogo[:ZPARAMeter]:NUMBER<x>:MODE?
<x> = 1 to 4

Example :GONOGO:ZPARAMETER:NUMBER1:
MODE PARAMETER
:GONOGO:ZPARAMETER:NUMBER1:MODE?
-> :GONOGO:ZPARAMETER:NUMBER1:
MODE PARAMETER

:GONogo[:ZPARAMeter]:NUMBER<x>:PARame ter?

Function Queries all the parameter settings of the specified condition.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:
PARame ter?
<x> = 1 to 4

5.13 GONogo Group

:GONogo[:ZPARAMeter]:NUMBer<x>:PARame ter:ITEM

Function Sets or queries the specified waveform parameter's item for parameter determination.

Syntax :GONogo[:ZPARAMeter]:NUMBer<x>:
PARAMeter:ITEM {<Parameter>}
:GONogo[:ZPARAMeter]:NUMBer<x>:
PARAMeter:ITEM?
<x> = 1 to 4
<Parameter> = {AMPLitude|AVERAge|AVGFreq
|AVGPeriod|BWIDth|DELAy|DT|DUTYcycle|ENU
Mber|FALL|FREQuency|HIGH|LOW|MAXimum|M
INimum|NOVershoot|NWIDth|PERiod|PNUMber
|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEV
iation|TY1Integ|TY2Integ|V1|V2}

Example :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:ITEM AVERAGE
:GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:ITEM?
-> :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:ITEM AVERAGE

:GONogo[:ZPARAMeter]:NUMBer<x>:PARame ter:LIMit

Function Sets or queries the specified waveform parameter's upper and lower limits for parameter determination.

Syntax :GONogo[:ZPARAMeter]:NUMBer<x>:
PARAMeter:LIMit {<NRf>}
:GONogo[:ZPARAMeter]:NUMBer<x>:
PARAMeter:LIMit?
<x> = 1 to 4
<NRf> = See the Features Guide for this information.

Example :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:LIMIT 0,1
:GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:LIMIT?
-> :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:LIMIT 1.000E+00,0.000E+00

:GONogo[:ZPARAMeter]:NUMBer<x>:PARame ter:TRACe

Function Sets or queries the specified waveform parameter's source waveform for parameter determination.

Syntax :GONogo[:ZPARAMeter]:NUMBer<x>:
PARAMeter:TRACe {<NRf>|BIT<y>|
FFT<y>|MATH<y>|XY<y>}
:GONogo[:ZPARAMeter]:NUMBer<x>:
PARAMeter:TRACe?
<x> = 1 to 4
<NRf> = 1 to 4
BIT<y>'s <y> = 1 to 8
FFT<y>'s <y> = 1 or 2
MATH<y>'s <y> = 1 to 4
XY<y>'s <y> = 1 or 2

Example :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:TRACE 1
:GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:TRACE?
-> :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:TRACE 1

:GONogo[:ZPARAMeter]:NUMBer<x>:PARame ter:VALue?

Function Queries the measured value of the specified waveform parameter for parameter determination.

Syntax :GONogo[:ZPARAMeter]:NUMBer<x>:
PARAMeter:VALue?
<x> = 1 to 4

Example :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:VALUE?
-> :GONOGO:ZPARAMETER:NUMBER1:
PARAMETER:VALUE 1.98E-03

Description If the mode is off or the value cannot be measured, the instrument returns "NAN" (not a number).

:GONogo[:ZPARAMeter]:NUMBer<x>:POLYg on?

Function Queries all polygonal zone determination settings.

Syntax :GONogo[:ZPARAMeter]:NUMBer<x>:
POLYgon?
<x> = 1 to 4

:GONogo[:ZPARAMeter]:NUMBER<x>:POLYgon:HPOSition

Function Sets or queries the horizontal position used in polygonal zone determination.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:HPOSition {<NRf>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:HPOSition?
<x> = 1 to 4
<NRf> = -5 to 5 (div)

Example :GONOGO:ZPARAMETER:NUMBER1:POLYGON:
HPOSITION 1
:GONOGO:ZPARAMETER:NUMBER1:POLYGON:
HPOSITION?
-> :GONOGO:ZPARAMETER:NUMBER1:
POLYGON:HPOSITION 1.000E+00

Description When TRACe is XY<x>, <NRf> = -4 div to 4 div.

:GONogo[:ZPARAMeter]:NUMBER<x>:POLYgon:RANGe

Function Sets or queries the source window used in polygonal zone determination.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:RANGe {MAIN|Z1|Z2}
:GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:RANGe?
<x> = 1 to 4

Example :GONOGO:ZPARAMETER:NUMBER1:POLYGON:
RANGE MAIN
:GONOGO:ZPARAMETER:NUMBER1:POLYGON:
RANGE?
-> :GONOGO:ZPARAMETER:NUMBER1:
POLYGON:RANGE MAIN

:GONogo[:ZPARAMeter]:NUMBER<x>:POLYgon:TRACe

Function Sets or queries the source waveform used in polygonal zone determination.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:TRACe {<NRf>|MATH<y>|XY<y>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:TRACe?
<x> = 1 to 4
<NRf> = 1 to 4
MATH<y>'s <y> = 1 to 4
XY<y>'s <y> = 1 to 2

Example :GONOGO:ZPARAMETER:NUMBER1:POLYGON:
TRACE 1
:GONOGO:ZPARAMETER:NUMBER1:POLYGON:
TRACE?
-> :GONOGO:ZPARAMETER:NUMBER1:
POLYGON:TRACE 1

:GONogo[:ZPARAMeter]:NUMBER<x>:POLYgon:VPOSition

Function Sets or queries the vertical position used in polygonal zone determination.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:VPOSition {<NRf>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:VPOSition?
<x> = 1 to 4
<NRf> = -4 to 4 (div)

Example :GONOGO:ZPARAMETER:NUMBER1:POLYGON:
VPOSITION 1
:GONOGO:ZPARAMETER:NUMBER1:POLYGON:
VPOSITION?
-> :GONOGO:ZPARAMETER:NUMBER1:
POLYGON:VPOSITION 1.000E+00

:GONogo[:ZPARAMeter]:NUMBER<x>:POLYgon:ZNUMber

Function Sets or queries the zone number used in polygonal zone determination.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:ZNUMber {<NRf>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
POLYgon:ZNUMber?
<x> = 1 to 4
<NRf> = 1 to 4

Example :GONOGO:ZPARAMETER:NUMBER1:POLYGON:
ZNUMBER 1
:GONOGO:ZPARAMETER:NUMBER1:POLYGON:
ZNUMBER?
-> :GONOGO:ZPARAMETER:NUMBER1:
POLYGON:ZNUMBER 1

:GONogo[:ZPARAMeter]:NUMBER<x>:RECTAng1e?

Function Queries all waveform rectangular zone determination settings.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle?
<x> = 1 to 4

5.13 GONogo Group

:GONogo[:ZPARAMeter]:NUMBER<x>:RECTangle:HORizontal

Function Sets or queries the horizontal position of the rectangle used in rectangular zone determination.

Syntax
:GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle:HORizontal {<NRf>,<NRf>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle:HORizontal?
<x> = 1 to 4
<NRf> = -5 to 5 (div)

Example
:GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:HORIZONTAL 1,2
:GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:HORIZONTAL?
-> :GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:HORIZONTAL 2.000E+00,
1.000E+00

Description When TRACE is XY<x>, <NRf> = -4 div to 4 div.

:GONogo[:ZPARAMeter]:NUMBER<x>:RECTangle:RANGE

Function Sets or queries the source window of the rectangle used in rectangular zone determination.

Syntax
:GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle:RANGE {MAIN|Z1|Z2}
:GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle:RANGE?
<x> = 1 to 4

Example
:GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:RANGE MAIN
:GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:RANGE?
-> :GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:RANGE MAIN

:GONogo[:ZPARAMeter]:NUMBER<x>:RECTangle:TRACE

Function Sets or queries the source waveform of the rectangle used in rectangular zone determination.

Syntax
:GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle:TRACE {<NRf>|MATH<y>|
XY<y>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle:TRACE?
<x> = 1 to 4
<NRf> = 1 to 4
MATH<y>'s <y> = 1 to 4
XY<y>'s <y> = 1 to 2

Example
:GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:TRACE 1
:GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:TRACE?
-> :GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:TRACE 1

:GONogo[:ZPARAMeter]:NUMBER<x>:RECTangle:VERTical

Function Sets or queries the vertical position of the rectangle used in rectangular zone determination.

Syntax
:GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle:VERTical {<NRf>,<NRf>}
:GONogo[:ZPARAMeter]:NUMBER<x>:
RECTangle:VERTical?
<x> = 1 to 4
<NRf> = -4 to 4 (div)

Example
:GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:VERTICAL 1,2
:GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:VERTICAL?
-> :GONOGO:ZPARAMETER:NUMBER1:
RECTANGLE:VERTICAL 2.000E+00,
1.000E+00

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE?

Function Queries all waveform zone determination settings.

Syntax
:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE?
<x> = 1 to 4

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:EDIT:EXIT

Function Exits from the waveform zone edit menu.

Syntax
:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:
EDIT:EXIT {QUIT|STORE}
<x> = 1 to 4

Example
:GONOGO:ZPARAMETER:NUMBER1:WAVE:
EDIT:EXIT STORE

Description When STORE is specified, the edited contents are saved, and the edit menu is closed.
When QUIT is specified, the edit menu is closed without saving the edited contents.

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:EDIT:NEW

Function Sets the base waveform in waveform zone editing.

Syntax
:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:
EDIT:NEW {<NRf>|MATH<y>}
<x> = 1 to 4
<NRf> = 1 to 4
MATH<y>'s <y> = 1 to 4

Example
:GONOGO:ZPARAMETER:NUMBER1:WAVE:
EDIT:NEW 1

Description When you finish editing, you need to use ":GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:EDIT:EXIT" to exit from the edit menu.

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:EDIT:PART

Function Executes partial editing in waveform zone editing.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:
EDIT:PART {<NRf>,<NRf>,<NRf>,<NRf>}
<x> = 1 to 4

<NRf> = -5.00 to 5.00 (div, T_Range1/T_Range2), -8.00 to 8.00 (div (difference), up and down)

Example :GONOGO:ZPARAMETER:NUMBER1:WAVE:
EDIT:PART -2.00,2.00,1.00,1.00

Description When you finish editing, you need to use ":GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:EDIT:EXIT" to exit from the edit menu.

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:EDIT:WHOLE

Function Sets the entire waveform zone in waveform zone editing.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:
EDIT:WHOLE {<NRf>,<NRf>,<NRf>,<NRf>}
<x> = 1 to 4

<NRf> = -5.00 to 5.00 (div, left and right), 0 to 8.00 (div, up and down)

Example :GONOGO:ZPARAMETER:NUMBER1:WAVE:
EDIT:WHOLE 0.50,0.50,1.00,1.00

Description When you finish editing, you need to use ":GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:EDIT:EXIT" to exit from the edit menu.

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:RANGE

Function Sets or queries the source window used in waveform zone determination.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:
RANGE {MAIN|Z1|Z2}
:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:
RANGE?
<x> = 1 to 4

Example :GONOGO:ZPARAMETER:NUMBER1:WAVE:
RANGE MAIN
:GONOGO:ZPARAMETER:NUMBER1:WAVE:
RANGE?
-> :GONOGO:ZPARAMETER:NUMBER1:WAVE:
RANGE MAIN

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:TRACE

Function Sets or queries the source waveform used in waveform zone determination.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:
TRACE {<NRf>|MATH<y>}
:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:
TRACE?
<x> = 1 to 4
<NRf> = 1 to 4
MATH<y>'s <y> = 1 to 4

Example :GONOGO:ZPARAMETER:NUMBER1:WAVE:
TRACE 1
:GONOGO:ZPARAMETER:NUMBER1:WAVE:
TRACE?
-> :GONOGO:ZPARAMETER:NUMBER1:WAVE:
TRACE 1

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:TRANGE

Function Sets or queries the determination period used in waveform zone determination.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:
TRANGE {<NRf>,<NRf>}
:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:
TRANGE?
<NRf> = -5 to 5 (div)

Example :GONOGO:ZPARAMETER:NUMBER1:WAVE:
TRANGE 1,2
:GONOGO:ZPARAMETER:NUMBER1:WAVE:
TRANGE?
-> :GONOGO:ZPARAMETER:NUMBER1:WAVE:
TRANGE 2.000E+00,1.000E+00

:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:ZNUMBER

Function Sets or queries the zone number used in waveform zone determination.

Syntax :GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:
ZNUMBER {<NRf>}
:GONogo[:ZPARAMeter]:NUMBER<x>:WAVE:
ZNUMBER?
<x> = 1 to 4
<NRf> = 1 to 4

Example :GONOGO:ZPARAMETER:NUMBER1:WAVE:
ZNUMBER 1
:GONOGO:ZPARAMETER:NUMBER1:WAVE:
ZNUMBER?
-> :GONOGO:ZPARAMETER:NUMBER1:WAVE:
ZNUMBER 1

5.14 HCOPy group

:HCOPy?

Function Queries all screen capture data output settings.
Syntax :HCOPy?

:HCOPy:ABORT

Function Stops data output and paper feeding.
Syntax :HCOPy:ABORT
Example :HCOPy:ABORT

:HCOPy:DIREction

Function Sets or queries the data output destination.
Syntax :HCOPy:DIREction {EXTPrinter|
PRINTER|NETPrint|FILE|MULTitarget}
:HCOPy:DIREction?
Example :HCOPy:DIRECTION EXTPRINTER
:HCOPy:DIRECTION?
-> :HCOPy:DIRECTION EXTPRINTER

:HCOPy:EXECute

Function Executes data output.
Syntax :HCOPy:EXECute
Example :HCOPy:EXECUTE

:HCOPy:EXTPrinter?

Function Queries all external printer output settings.
Syntax :HCOPy:EXTPrinter?

:HCOPy:EXTPrinter:MODE

Function Sets or queries the normal or hard copy setting
for outputting to the external printer.
Syntax :HCOPy:EXTPrinter:MODE {HARD|NORMAL}
:HCOPy:EXTPrinter:MODE?
Example :HCOPy:EXTPRINTER:MODE HARD
:HCOPy:EXTPRINTER:MODE?
-> :HCOPy:EXTPRINTER:MODE HARD

:HCOPy:EXTPrinter:TONE

Function Sets or queries the external printer output color.
Syntax :HCOPy:EXTPrinter:TONE {<Boolean>
:HCOPy:EXTPrinter:TONE?
Example :HCOPy:EXTPRINTER:TONE ON
:HCOPy:EXTPRINTER:TONE?
-> :HCOPy:EXTPRINTER:TONE 1

:HCOPy:MULTitarget?

Function Queries all multi target feature settings.
Syntax :HCOPy:MULTitarget?

:HCOPy:MULTitarget:EXTPrinter

Function Sets or queries whether the multi target feature
will send output to the external printer.
Syntax :HCOPy:MULTitarget:
EXTPrinter {<Boolean>
:HCOPy:MULTitarget:EXTPrinter?
Example :HCOPy:MULTITARGET:EXTPRINTER ON
:HCOPy:MULTITARGET:EXTPRINTER?
-> :HCOPy:MULTITARGET:EXTPRINTER 1

:HCOPy:MULTitarget:PRINTER

Function Sets or queries whether the multi target feature
will send output to the built-in printer.
Syntax :HCOPy:MULTitarget:
PRINTER {<Boolean>
:HCOPy:MULTitarget:PRINTER?
Example :HCOPy:MULTITARGET:PRINTER ON
:HCOPy:MULTITARGET:PRINTER?
-> :HCOPy:MULTITARGET:PRINTER 1

:HCOPy:MULTitarget:NETPrinter

Function Sets or queries whether the multi target feature
will send output to a network printer.
Syntax :HCOPy:MULTitarget:
NETPrinter {<Boolean>
:HCOPy:MULTitarget:NETPrinter?
Example :HCOPy:MULTITARGET:NETPRINTER ON
:HCOPy:MULTITARGET:NETPRINTER?
-> :HCOPy:MULTITARGET:NETPRINTER 1

:HCOPy:MULTitarget:FILE

Function Sets or queries whether the multi target feature
will send output to a file.
Syntax :HCOPy:MULTitarget:FILE {<Boolean>
:HCOPy:MULTitarget:FILE?
Example :HCOPy:MULTITARGET:FILE ON
:HCOPy:MULTITARGET:FILE?
-> :HCOPy:MULTITARGET:FILE 1

:HCOPY:MULTitarget:WAVEform

Function Sets or queries whether the multi target feature will send output to a waveform file.

Syntax :HCOPY:MULTitarget:
WAVEform{<Boolean>}
:HCOPY:MULTitarget:WAVEform?

Example :HCOPY:MULTITARGET:WAVEFORM ON
:HCOPY:MULTITARGET:WAVEFORM?
-> :HCOPY:MULTITARGET:WAVEFORM 1

:HCOPY:NETPrint?

Function Queries all network printer output settings.

Syntax :HCOPY:NETPrint?

:HCOPY:NETPrint:MODE

Function Sets or queries the normal or hard copy setting for outputting to the network printer.

Syntax :HCOPY:NETPrint:MODE {HARD|NORMal}
:HCOPY:NETPrint:MODE?

Example :HCOPY:NETPRINT:MODE HARD
:HCOPY:NETPRINT:MODE?
-> :HCOPY:NETPRINT:MODE HARD

:HCOPY:NETPrint:TONE

Function Sets or queries the network printer output color.

Syntax :HCOPY:NETPrint:TONE {<Boolean>}
:HCOPY:NETPrint:TONE?

Example :HCOPY:NETPRINT:TONE ON
:HCOPY:NETPRINT:TONE?
-> :HCOPY:NETPRINT:TONE 1

:HCOPY:NETPrint:TYPE

Function Sets or queries the network printer output command type.

Syntax :HCOPY:NETPrint:TYPE {HINKjet|
HLASer}
:HCOPY:NETPrint:TYPE?

Example :HCOPY:NETPRINT:TYPE EINKJET
:HCOPY:NETPRINT:TYPE?
-> :HCOPY:NETPRINT:TYPE EINKJET

:HCOPY:PRINter?

Function Queries all built-in printer output settings.

Syntax :HCOPY:PRINter?

:HCOPY:PRINter:MODE

Function Sets or queries the short or hard copy setting for outputting to the built-in printer.

Syntax :HCOPY:PRINter:MODE {HARD|SHORT}
:HCOPY:PRINter:MODE?

Example :HCOPY:PRINTER:MODE HARD
:HCOPY:PRINTER:MODE?
-> :HCOPY:PRINTER:MODE HARD

:HCOPY:PRINter:REPort

Function Sets or queries whether additional information is to be output to the built-in printer.

Syntax :HCOPY:PRINter:REPort {<Boolean>}
:HCOPY:PRINter:REPort?

Example :HCOPY:PRINTER:REPORT ON
:HCOPY:PRINTER:REPORT?
-> :HCOPY:PRINTER:REPORT 1

5.15 HISTory Group

:HISTory?

Function Queries all history feature settings.
Syntax :HISTory?

:HISTory:AVERage

Function Sets or queries the history waveform highlight display mode.
Syntax :HISTory:AVERage {<Boolean>
:HISTory:AVERage?
Example :HISTORY:AVERAGE ON
:HISTORY:AVERAGE?
-> :HISTORY:AVERAGE 1

:HISTory:DISPlay

Function Sets or queries the history record start and end numbers that will be displayed.
Syntax :HISTory:DISPlay {<Nrf>,<Nrf>}
:HISTory:DISPlay?
<Nrf> = See the Features Guide for this information.
Example :HISTORY:DISPLAY 0,-100
:HISTORY:DISPLAY?
-> :HISTORY:DISPLAY 0,-100

:HISTory:DMODE

Function Sets or queries the history waveform display mode.
Syntax :HISTory:DMODE {ALL|COLOR|INTensity|ONE}
:HISTory:DMODE?
Example :HISTORY:DMODE ALL
:HISTORY:DMODE?
-> :HISTORY:DMODE ALL

:HISTory:RECORD

Function Sets or queries the history waveform source record.
Syntax :HISTory:RECORD {<Nrf>|MINimum}
:HISTory:RECORD?
<Nrf> = See the Features Guide for this information.
Example :HISTORY:RECORD 0
:HISTORY:RECORD?
-> :HISTORY:RECORD 0
Description Specify "MINimum" to specify the minimum record number.

:HISTory:REPLay?

Function Queries all history waveform replay settings.
Syntax :HISTory:REPLay?

:HISTory:REPLay:JUMP

Function Moves to the specified record number in the history waveform.
Syntax :HISTory:REPLay:JUMP {MAXimum|MINimum}
Example :HISTORY:REPLAY:JUMP MAXIMUM

:HISTory:REPLay:SPEEd

Function Sets or queries the history waveform replay speed.
Syntax :HISTory:REPLay:SPEEd {<Nrf>|PER3|PER10|PER30|PER60}
:HISTory:REPLay:SPEEd?
<Nrf> = 1, 3, 10
Example :HISTORY:REPLAY:SPEED 1
:HISTORY:REPLAY:SPEED?
-> :HISTORY:REPLAY:SPEED 1

:HISTory:REPLay:START

Function Starts replaying the history waveform in the specified direction.
Syntax :HISTory:REPLay:START {MAXimum|MINimum}
Example :HISTORY:REPLAY:START MAXIMUM

:HISTory:REPLay:STOP

Function Stops replaying the history waveform.
Syntax :HISTory:REPLay:STOP
Example :HISTORY:REPLAY:STOP

:HISTory:SEARCh?

Function Queries all history waveform search settings.
Syntax :HISTory:SEARCh?

:HISTory[:SEARCh]:ABORT

Function Aborts the search.
Syntax :HISTory[:SEARCh]:ABORT
Example :HISTORY:SEARCH:ABORT

:HISTory[:SEARch]:EXECute

Function Executes a search. This is an overlap command.

Syntax :HISTory[:SEARch]:EXECute

Example :HISTORY:SEARCH:EXECUTE

:HISTory[:SEARch]:LOGic

Function Sets or queries the history waveform search logic.

Syntax :HISTory[:SEARch]:LOGic {AND|OR|SIMPlE}

:HISTory[:SEARch]:LOGic?

Example :HISTORY:SEARCH:LOGIC AND

:HISTORY:SEARCH:LOGIC?

-> :HISTORY:SEARCH:LOGIC AND

:HISTory[:SEARch]:NUMBer<x>?

Function Queries all search condition settings.

Syntax :HISTory[:SEARch]:NUMBer<x>?

<x> = 1 to 4

:HISTory[:SEARch]:NUMBer<x>:CONDition

Function Sets or queries the reference condition of the specified search condition.

Syntax :HISTory[:SEARch]:NUMBer<x>:

CONDition {IN|OUT}

:HISTory[:SEARch]:NUMBer<x>:

CONDition?

<x> = 1 to 4

Example :HISTORY:SEARCH:NUMBER1:CONDITION IN

:HISTORY:SEARCH:NUMBER1:CONDITION?

-> :HISTORY:SEARCH:NUMBER1:

CONDITION IN

:HISTory[:SEARch]:NUMBer<x>:MODE

Function Sets or queries the mode of the specified search condition.

Syntax :HISTory[:SEARch]:NUMBer<x>:

MODE {OFF|PARAMeter|POLYgon|

RECTangle|WAVE}

:HISTory[:SEARch]:NUMBer<x>:MODE?

<x> = 1 to 4

Example :HISTORY:SEARCH:NUMBER1:

MODE PARAMETER

:HISTORY:SEARCH:NUMBER1:MODE?

-> :HISTORY:SEARCH:NUMBER1:

MODE PARAMETER

:HISTory[:SEARch]:NUMBer<x>:PARAMeter?

Function Queries all parameter search settings.

Syntax :HISTory[:SEARch]:NUMBer<x>:

PARAMeter?

<x> = 1 to 4

:HISTory[:SEARch]:NUMBer<x>:PARAMeter:**ITEM**

Function Sets or queries the specified waveform parameter's item for parameter searching.

Syntax :HISTory[:SEARch]:NUMBer<x>:

PARAMeter:ITEM <Parameter>

:HISTory[:SEARch]:NUMBer<x>:

PARAMeter:ITEM?

<x> = 1 to 4

<Parameter> = {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth|DElay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDth|PERiod|PNUMBER|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEVIation|TY1Integ|TY2Integ|V1|V2}

Example :HISTORY:SEARCH:NUMBER1:PARAMETER:

ITEM AVERAGE

:HISTORY:SEARCH:NUMBER1:PARAMETER:

ITEM?

-> :HISTORY:SEARCH:NUMBER1:

PARAMETER:ITEM AVERAGE

:HISTory[:SEARch]:NUMBer<x>:PARAMeter:**LIMit**

Function Sets or queries the specified waveform parameter's upper and lower limits for parameter searching.

Syntax :HISTory[:SEARch]:NUMBer<x>:

PARAMeter:LIMit {<NRf>}

:HISTory[:SEARch]:NUMBer<x>:

PARAMeter:LIMit?

<x> = 1 to 4

<NRf> = See the Features Guide for this information.

Example :HISTORY:SEARCH:NUMBER1:PARAMETER:

LIMIT 0,1

:HISTORY:SEARCH:NUMBER1:PARAMETER:

LIMIT?

-> :HISTORY:SEARCH:NUMBER1:

PARAMETER:LIMIT 1.000E+00,0.000E+00

5.15 HISTory Group

:HISTory[:SEARCh]:NUMBer<x>:PARAmeter:TRACe

Function Sets or queries the specified waveform parameter's source waveform for parameter searching.

Syntax :HISTory[:SEARCh]:NUMBer<x>:
PARAmeter:TRACe {<NRf>|BIT<y>|
FFT<y>|MATH<y>|XY<y>}
:HISTory[:SEARCh]:NUMBer<x>:
PARAmeter:TRACe?

<x> = 1 to 4

<NRf> = 1 to 4

BIT<y>'s <y> = 1 to 8

FFT<y>'s <y> = 1 to 2

MATH<y>'s <y> = 1 to 4

XY<y>'s <y> = 1 to 2

Example :HISTORY:SEARCH:NUMBER1:PARAMETER:
TRACE 1
:HISTORY:SEARCH:NUMBER1:PARAMETER:
TRACE?
-> :HISTORY:SEARCH:NUMBER1:
PARAMETER:TRACE 1

:HISTory[:SEARCh]:NUMBer<x>:PARAmeter:VALue?

Function Queries the measured value of the specified waveform parameter for parameter searching.

Syntax :HISTory[:SEARCh]:NUMBer<x>:
PARAmeter:VALue?
<x> = 1 to 4

Example :HISTORY:SEARCH:NUMBER1:PARAMETER:
VALUE?
-> :HISTORY:SEARCH:NUMBER1:
PARAMETER:VALUE 1.98E-03

Description If the mode is off or the value cannot be measured, the instrument returns "NAN" (not a number).

:HISTory[:SEARCh]:NUMBer<x>:POLYgon?

Function Queries all polygonal zone search settings.

Syntax :HISTory[:SEARCh]:NUMBer<x>:POLYgon?
<x> = 1 to 4

:HISTory[:SEARCh]:NUMBer<x>:POLYgon:HP OSition

Function Sets or queries the horizontal position used in polygonal zone searching.

Syntax :HISTory[:SEARCh]:NUMBer<x>:POLYgon:
HPOSITION {<NRf>}
:HISTory[:SEARCh]:NUMBer<x>:POLYgon:
HPOSITION?
<x> = 1 to 4
<NRf> = -5 to 5 (div)

Example :HISTORY:SEARCH:NUMBER1:POLYGON:
HPOSITION 1
:HISTORY:SEARCH:NUMBER1:POLYGON:
HPOSITION?
-> :HISTORY:SEARCH:NUMBER1:POLYGON:
HPOSITION 1.000E+00

Description When TRACe is XY<x>, <NRf> = -4 div to 4 div.

:HISTory[:SEARCh]:NUMBer<x>:POLYgon:RANGe

Function Sets or queries the source window used in polygonal zone searching.

Syntax :HISTory[:SEARCh]:NUMBer<x>:POLYgon:
RANGe {MAIN|Z1|Z2}
:HISTory[:SEARCh]:NUMBer<x>:POLYgon:
RANGe?
<x> = 1 to 4

Example :HISTORY:SEARCH:NUMBER1:POLYGON:
RANGE MAIN
:HISTORY:SEARCH:NUMBER1:POLYGON:
RANGE?
-> :HISTORY:SEARCH:NUMBER1:POLYGON:
RANGE MAIN

:HISTory[:SEARCh]:NUMBer<x>:POLYgon:TRACe

Function Sets or queries the source waveform used in polygonal zone searching.

Syntax :HISTory[:SEARCh]:NUMBer<x>:POLYgon:
TRACE {<NRf>|MATH<y>|XY<y>}
:HISTory[:SEARCh]:NUMBer<x>:POLYgon:
TRACE?
<x> = 1 to 4
<NRf> = 1 to 4
MATH<y>'s <y> = 1 to 4
XY<y>'s <y> = 1 to 2

Example :HISTORY:SEARCH:NUMBER1:POLYGON:
TRACE 1
:HISTORY:SEARCH:NUMBER1:POLYGON:
TRACE?
-> :HISTORY:SEARCH:NUMBER1:POLYGON:
TRACE 1

**:HISTory[:SEARCh]:NUMBer<x>:POLYgon:VP
OSition**

Function Sets or queries the vertical position used in polygonal zone searching.

Syntax :HISTory[:SEARCh]:NUMBer<x>:POLYgon:
VPOStion {<NRf>}
:HISTory[:SEARCh]:NUMBer<x>:POLYgon:
VPOStion?
<x> = 1 to 4
<NRf> = -4 to 4 (div)

Example :HISTORY:SEARCH:NUMBER1:POLYGON:
VPOSITION 1
:HISTORY:SEARCH:NUMBER1:POLYGON:
VPOSITION?
-> :HISTORY:SEARCH:NUMBER1:POLYGON:
VPOSITION 1.000E+00

**:HISTory[:SEARCh]:NUMBer<x>:POLYgon:ZN
UMber**

Function Sets or queries the zone number used in polygonal zone searching.

Syntax :HISTory[:SEARCh]:NUMBer<x>:POLYgon:
ZNUmber {<NRf>}
:HISTory[:SEARCh]:NUMBer<x>:POLYgon:
ZNUmber?
<x> = 1 to 4
<NRf> = 1 to 4

Example :HISTORY:SEARCH:NUMBER1:POLYGON:
ZNUMBER 1
:HISTORY:SEARCH:NUMBER1:POLYGON:
ZNUMBER?
-> :HISTORY:SEARCH:NUMBER1:POLYGON:
ZNUMBER 1

:HISTory[:SEARCh]:NUMBer<x>:RECTangle?

Function Queries all waveform rectangular zone search settings.

Syntax :HISTory[:SEARCh]:NUMBer<x>:
RECTangle?
<x> = 1 to 4

**:HISTory[:SEARCh]:NUMBer<x>:RECTangle:
HORizontal**

Function Sets or queries the horizontal position of the rectangle used in rectangular zone searching.

Syntax :HISTory[:SEARCh]:NUMBer<x>:
RECTangle:HORizontal {<NRf>,<NRf>}
:HISTory[:SEARCh]:NUMBer<x>:
RECTangle:HORizontal?
<x> = 1 to 4
<NRf> = -5 to 5 (div)

Example :HISTORY:SEARCH:NUMBER1:RECTANGLE:
HORIZONTAL 1,2
:HISTORY:SEARCH:NUMBER1:RECTANGLE:
HORIZONTAL?
-> :HISTORY:SEARCH:NUMBER1:
RECTANGLE:HORIZONTAL 2.000E+00,
1.000E+00

Description When TRACE is XY<x>, <NRf> = -4 div to 4 div.

**:HISTory[:SEARCh]:NUMBer<x>:RECTangle:
RANGE**

Function Sets or queries the source window of the rectangle used in rectangular zone searching.

Syntax :HISTory[:SEARCh]:NUMBer<x>:
RECTangle:RANGE {MAIN|Z1|Z2}
:HISTory[:SEARCh]:NUMBer<x>:
RECTangle:RANGE?
<x> = 1 to 4

Example :HISTORY:SEARCH:NUMBER1:RECTANGLE:
RANGE MAIN
:HISTORY:SEARCH:NUMBER1:RECTANGLE:
RANGE?
-> :HISTORY:SEARCH:NUMBER1:
RECTANGLE:RANGE MAIN

**:HISTory[:SEARCh]:NUMBer<x>:RECTangle:
TRACE**

Function Sets or queries the source waveform of the rectangle used in rectangular zone searching.

Syntax :HISTory[:SEARCh]:NUMBer<x>:
RECTangle:TRACE {<NRf>|MATH<y>|
XY<y>}
:HISTory[:SEARCh]:NUMBer<x>:
RECTangle:TRACE?
<x> = 1 to 4
<NRf> = 1 to 4
MATH<y>'s <y> = 1 to 4
XY<y>'s <y> = 1 to 2

Example :HISTORY:SEARCH:NUMBER1:RECTANGLE:
TRACE 1
:HISTORY:SEARCH:NUMBER1:RECTANGLE:
TRACE?
-> :HISTORY:SEARCH:NUMBER1:
RECTANGLE:TRACE 1

5.15 HISTory Group

:HISTory[:SEARCh]:NUMBER<x>:RECTangle:VERTical

Function Sets or queries the vertical position of the rectangle used in rectangular zone searching.

Syntax :HISTory[:SEARCh]:NUMBER<x>:
RECTangle:VERTical {<Nrf>,<Nrf>}
:HISTory[:SEARCh]:NUMBER<x>:
RECTangle:VERTical?
<x> = 1 to 4
<Nrf> = -4 to 4 (div)

Example :HISTORY:SEARCH:NUMBER1:RECTANGLE:
VERTICAL 1,2
:HISTORY:SEARCH:NUMBER1:RECTANGLE:
VERTICAL?
-> :HISTORY:SEARCH:NUMBER1:
RECTANGLE:VERTICAL 2.000E+00,
1.000E+00

:HISTory[:SEARCh]:NUMBER<x>:WAVE?

Function Queries all waveform zone search settings.

Syntax :HISTory[:SEARCh]:NUMBER<x>:WAVE?
<x> = 1 to 4

:HISTory[:SEARCh]:NUMBER<x>:WAVE:RANGE

Function Sets or queries the source window used in waveform zone searching.

Syntax :HISTory[:SEARCh]:NUMBER<x>:WAVE:
RANGE {MAIN|Z1|Z2}
:HISTory[:SEARCh]:NUMBER<x>:WAVE:
RANGE?
<x> = 1 to 4

Example :HISTORY:SEARCH:NUMBER1:WAVE:
RANGE MAIN
:HISTORY:SEARCH:NUMBER1:WAVE:RANGE?
-> :HISTORY:SEARCH:NUMBER1:WAVE:
RANGE MAIN

:HISTory[:SEARCh]:NUMBER<x>:WAVE:TRACe

Function Sets or queries the source waveform used in waveform zone searching.

Syntax :HISTory[:SEARCh]:NUMBER<x>:WAVE:
TRACe {<Nrf>|MATH<y>}
:HISTory[:SEARCh]:NUMBER<x>:WAVE:
TRACe?
<x> = 1 to 4
<Nrf> = 1 to 4
MATH<y>'s <y> = 1 to 4

Example :HISTORY:SEARCH:NUMBER1:WAVE:TRACE 1
:HISTORY:SEARCH:NUMBER1:WAVE:TRACE?
-> :HISTORY:SEARCH:NUMBER1:WAVE:
TRACE 1

:HISTory[:SEARCh]:NUMBER<x>:WAVE:TRAN ge

Function Sets or queries the determination period used in waveform zone searching.

Syntax :HISTory[:SEARCh]:NUMBER<x>:WAVE:
TRAN ge {<Nrf>,<Nrf>}
:HISTory[:SEARCh]:NUMBER<x>:WAVE:
TRAN ge?
<Nrf> = -5 to 5 (div)

Example :HISTORY:SEARCH:NUMBER1:WAVE:
TRAN GE 1,2
:HISTORY:SEARCH:NUMBER1:WAVE:TRAN GE?
-> :HISTORY:SEARCH:NUMBER1:WAVE:
TRAN GE 2.000E+00,1.000E+00

:HISTory[:SEARCh]:NUMBER<x>:WAVE:ZNUMb er

Function Sets or queries the zone number used in waveform zone searching.

Syntax :HISTory[:SEARCh]:NUMBER<x>:WAVE:
ZNUMBER {<Nrf>}
:HISTory[:SEARCh]:NUMBER<x>:WAVE:
ZNUMBER?
<x> = 1 to 4
<Nrf> = 1 to 4

Example :HISTORY:SEARCH:NUMBER1:WAVE:
ZNUMBER 1
:HISTORY:SEARCH:NUMBER1:WAVE:
ZNUMBER?
-> :HISTORY:SEARCH:NUMBER1:WAVE:
ZNUMBER 1

:HISTory[:SEARCh]:RESet

Function Resets the history waveform search conditions.

Syntax :HISTory[:SEARCh]:RESet

Example :HISTORY:SEARCH:RESET

:HISTory[:SEARCh]:SIMPlE?

Function Queries all simple searching settings.

Syntax :HISTory[:SEARCh]:SIMPlE?

:HISTory[:SEARCh]:SIMPlE:HORizontal

Function Sets or queries the horizontal position of the rectangle to use in simple searching.

Syntax :HISTory[:SEARCh]:SIMPlE:
HORizontal {<Nrf>,<Nrf>}
:HISTory[:SEARCh]:SIMPlE:HORizontal?
<Nrf> = -5 to 5 (div)

Example :HISTORY:SEARCH:SIMPLE:HORIZONTAL 1,2
:HISTORY:SEARCH:SIMPLE:HORIZONTAL?
-> :HISTORY:SEARCH:SIMPLE:
HORIZONTAL 2.000E+00,1.000E+00

:HISTory[:SEARCh]:SIMPlE:RANGe

Function Sets or queries the target window of the rectangle to use in simple searching.

Syntax :HISTory[:SEARCh]:SIMPlE:
RANGe {MAIN|Z1|Z2}
:HISTory[:SEARCh]:SIMPlE:RANGe?

Example :HISTORY:SEARCH:SIMPLE:RANGE MAIN
:HISTORY:SEARCH:SIMPLE:RANGE?
-> :HISTORY:SEARCH:SIMPLE:RANGE MAIN

:HISTory[:SEARCh]:SIMPlE:TRACe

Function Sets or queries the source trace of the rectangle to use in simple searching.

Syntax :HISTory[:SEARCh]:SIMPlE:
TRACe {<NRf>|MATH<x>|XY<x>}
:HISTory[:SEARCh]:SIMPlE:TRACe?
<NRf> = 1 to 4
MATH<x>'s <x> = 1 to 4
XY<x>'s <x> = 1, 2

Example :HISTORY:SEARCH:SIMPLE:TRACE 1
:HISTORY:SEARCH:SIMPLE:TRACE?
-> :HISTORY:SEARCH:SIMPLE:TRACE 1

:HISTory[:SEARCh]:SIMPlE:VERTical

Function Sets or queries the vertical position of the rectangle to use in simple searching.

Syntax :HISTory[:SEARCh]:SIMPlE:
VERTical {<NRf>,<NRf>}
:HISTory[:SEARCh]:SIMPlE:VERTical?
<NRf> = -4 to 4 (div)

Example :HISTORY:SEARCH:SIMPLE:VERTICAL 1,2
:HISTORY:SEARCH:SIMPLE:VERTICAL?
-> :HISTORY:SEARCH:SIMPLE:
VERTICAL 2.000E+00,1.000E+00

:HISTory:TIME?

Function Queries the time of the source record number.

Syntax :HISTory:TIME? {<NRf>|MINimum}

Example :HISTORY:TIME? -100
-> :HISTORY:TIME "-100 10:20:30.400"

Description Specify "MINimum" to specify the minimum record number.

5.16 IMAGE Group

:IMAGE?

Function Queries all screen capture data output settings.
Syntax :IMAGE?

:IMAGE:ABORT

Function Aborts saving screen capture data to the storage device.
Syntax :IMAGE:ABORT
Example :IMAGE:ABORT

:IMAGE:BACKground

Function Sets or queries the screen capture background.
Syntax :IMAGE:BACKground {NORMAL|TRANSPARENT}
:IMAGE:BACKground?
Example :IMAGE:BACKGROUND NORMAL
:IMAGE:BACKGROUND?
-> :IMAGE:BACKGROUND NORMAL

:IMAGE:EXECute

Function Saves the screen capture data to the storage device.
Syntax :IMAGE:EXECute
Example :IMAGE:EXECUTE

:IMAGE:FORMat

Function Sets or queries the screen capture output format.
Syntax :IMAGE:FORMat {BMP|JPEG|PNG}
:IMAGE:FORMat?
Example :IMAGE:FORMAT BMP
:IMAGE:FORMAT?
-> :IMAGE:FORMAT BMP

:IMAGE:INformation

Function Sets or queries whether setting information is included in screen capture data.
Syntax :IMAGE:INformation {<Boolean>}
:IMAGE:INformation?
Example :IMAGE:INFORMATION ON
:IMAGE:INFORMATION?
-> :IMAGE:INFORMATION 1

:IMAGE:MODE

Function Sets or queries the screen capture output mode.
Syntax IMAGE:MODE {HARD1|HARD2|WIDE}
:IMAGE:MODE?
Example :IMAGE:MODE HARD1
:IMAGE:MODE?
-> :IMAGE:MODE HARD1

:IMAGE:SAVE?

Function Queries all file output settings.
Syntax :IMAGE:SAVE?

:IMAGE:SAVE:ANAMing

Function Sets or queries the on/off state of the auto naming feature for saving files.
Syntax :IMAGE:SAVE:ANAMing {DATE|DATE2|NUMBERING|OFF}
:IMAGE:SAVE:ANAMing?
Example :IMAGE:SAVE:ANAMING DATE
:IMAGE:SAVE:ANAMING?
-> :IMAGE:SAVE:ANAMING DATE

:IMAGE:SAVE:CDIRECTory

Function Changes the file directory.
Syntax :IMAGE:SAVE:CDIRECTory {<String>}
<String> = See the Features Guide for this information.
Example :IMAGE:SAVE:CDIRECTORY "ABC"

:IMAGE:SAVE:DRIVE

Function Sets the storage device that the file will be created in.
Syntax :IMAGE:SAVE:DRIVE {FLASHmem|NETWork|USB,<Nrf>}
<Nrf> = 0 to 1
Example :IMAGE:SAVE:DRIVE FLASHMEM
Description The <Nrf> for USB can be omitted if the drive is not partitioned or divided by LUN.

:IMAGE:SAVE:NAME

Function Sets or queries the name of the file to be created.
Syntax :IMAGE:SAVE:NAME {<Filename>}
:IMAGE:SAVE:NAME?
Example :IMAGE:SAVE:NAME "DISP_1"
:IMAGE:SAVE:NAME?
-> :IMAGE:SAVE:NAME "DISP_1"

:IMAGE:SEND?

Function Queries the screen capture data value.

Syntax :IMAGE:SEND?

Example :IMAGE:SEND?

-> :IMAGE:

SEND #9 (9-digit number of bytes)(data

byte sequence)(block data)

Description For details on <block data>, see page 4-10.

:IMAGE:TONE

Function Sets or queries the color tone of the screen capture data that will be saved.

Syntax :IMAGE:TONE {COLor|GRAY|OFF|REVerse}

:IMAGE:TONE?

Example :IMAGE:TONE COLOR

:IMAGE:TONE?

-> :IMAGE:TONE COLOR

:IMAGE:WGRadation

Function Sets or queries whether or not gradation is applied to the waveform portion of the screen image data to be output.

Syntax :IMAGE:WGRadation {<boolean>}

:IMAGE:WGRadation?

Example :IMAGE:WGRADATION ON

:IMAGE:WGRADATION?

-> :IMAGE:WGRADATION 1

Description This command is valid only when :IMAGE:TONE REVerse is selected.

5.17 INITialize Group

:INITialize:EXECute

Function Executes initialization.

Syntax :INITialize:EXECute

Example :INITIALIZE:EXECUTE

:INITialize:UNDO

Function Undoes the initialization that was executed.

Syntax :INITialize:UNDO

Example :INITIALIZE:UNDO

5.18 LOGic group

The commands in this group cannot be used on models without logic input.

:LOGic?

Function Queries all logic input waveform settings.

Syntax :LOGic?

:LOGic:MODE

Function Sets or queries the logic input on/off state.

Syntax :LOGic:MODE {<Boolean>}

:LOGic:MODE?

Example :LOGIC:MODE ON

:LOGIC:MODE?

-> :LOGIC:MODE 1

:LOGic:PODA?

Function Queries all logic input settings.

Syntax :LOGic:PODA?

:LOGic[:PODA]:BITOrder

Function Sets or queries bit order of the logic input.

Syntax :LOGic[:PODA]:BITOrder {<String>}

:LOGic[:PODA]:BITOrder?

<String> = Combination of "BIT0" to "BIT7" and "BUS" (35 characters)

Example :LOGIC:PODA:

BITORDER "BIT0BIT1BIT2BIT3BIT4BIT5BIT6BIT7BUS"

:LOGIC:PODA:BITORDER?

-> :LOGIC:PODA:

BITORDER "BIT0BIT1BIT2BIT3BIT4BIT5BIT6BIT7BUS"

:LOGic[:PODA]:BITSetup:ALL:DISPlay

Function Turns on or off all bit displays of the specified pod of the logic input.

Syntax :LOGic[:PODA]:BITSetup:ALL:

DISPlay {<Boolean>}

:LOGic[:PODA]:BITSetup:ALL:DISPlay?

Example :LOGIC:PODA:ALL:DISPLAY ON

:LOGIC:PODA:ALL:DISPLAY?

-> :LOGIC:PODA:ALL:DISPLAY 1

:LOGic[:PODA]:BITSetup:ALL:LEVel

Function Sets or queries the user-defined threshold level of the specified pod of the logic input.

Syntax :LOGic[:PODA]:BITSetup:ALL:

LEVel {<Voltage>}

:LOGic[:PODA]:BITSetup:ALL:LEVel?

<Voltage> = -10 V to 10 V, 0.1 V steps

Example :LOGIC:PODA:BITSETUP:ALL:LEVEL 1V

:LOGIC:PODA:BITSETUP:ALL:LEVEL?

-> :LOGIC:PODA:BITSETUP:ALL:

LEVEL 1.0E+00

Description The selectable <voltage> range varies depending on the logic probe that you use as follows.

- Model 701988: -40 V to 40 V (0.05 V steps)
- Model 701989: -6 V to 6 V (0.05 V steps)

:LOGic[:PODA]:BITSetup:ALL:TYPE

Function Selects the threshold level of the specified pod of the logic input.

Syntax :LOGic[:PODA]:BITSetup:ALL:

TYPE {CMOS1|CMOS2|CMOS3|CMOS5|ECL}

Example :LOGIC:PODA:BITSETUP:ALL:TYPE CMOS1

:LOGic[:PODA]:BITSetup:BIT<x>?

Function Queries all settings of a bit of the specified pod of the logic input.

Syntax :LOGic[:PODA]:BITSetup:BIT<x>?

<x> = 1 to 8

:LOGic[:PODA]:BITSetup:BIT<x>:DISPlay

Function Turns on or off all bit displays of the specified pod of the logic input.

Syntax :LOGic[:PODA]:BITSetup:BIT<x>:

DISPlay {<Boolean>}

:LOGic[:PODA]:BITSetup:BIT<x>:

DISPlay?

<x> = 1 to 8

Example :LOGIC:PODA:BITSETUP:BIT1:DISPLAY ON

:LOGIC:PODA:BITSETUP:BIT1:DISPLAY?

-> :LOGIC:PODA:BITSETUP:BIT1:

DISPLAY 1

5.18 LOGic group

:LOGic[:PODA]:BITSetup:BIT<x>:LABel

Function Sets or queries the label of a bit of the specified pod of the logic input.

Syntax :LOGic[:PODA]:BITSetup:BIT<x>:
LABel {<String>}
:LOGic[:PODA]:BITSetup:BIT<x>:LABel?
<x> = 1 to 8
<String> = Up to 8 characters

Example :LOGIC:PODA:BITSETUP:BIT1:
LABEL "NO _ 1"
:LOGIC:PODA:BITSETUP:BIT1:LABEL?
-> :LOGIC:PODA:BITSETUP:BIT1:
LABEL "NO _ 1"

:LOGic[:PODA]:BITSetup:BIT<x>:LEVel

Function Sets or queries the user-defined threshold level of a bit of the specified pod of the logic input.

Syntax :LOGic[:PODA]:BITSetup:BIT<x>:
LEVel {<Voltage>}
:LOGic[:PODA]:BITSetup:BIT<x>:LEVel?
<x> = 1 to 8
<Voltage> = -10 V to 10 V, 0.1 V steps

Example :LOGIC:PODA:BITSETUP:BIT1:LEVEL 1V
:LOGIC:PODA:BITSETUP:BIT1:LEVEL?
-> :LOGIC:PODA:BITSETUP:BIT1:
LEVEL 1.0E+00

Description The selectable <voltage> range varies depending on the logic probe that you use as follows.

- Model 701988: -40 V to 40 V (0.05 V steps)
- Model 701989: -6 V to 6 V (0.05 V steps)

:LOGic[:PODA]:BITSetup:BIT<x>:TYPE

Function Selects the threshold level of a bit of the specified pod of the logic input.

Syntax :LOGic[:PODA]:BITSetup:BIT<x>:
TYPE {CMOS1|CMOS2|CMOS3|CMOS5|ECL}
<x> = 1 to 8

Example :LOGIC:PODA:BITSETUP:BIT1:TYPE CMOS1

:LOGic[:PODA]:BUS?

Function Queries all logic input bus settings.

Syntax :LOGic[:PODA]:BUS?

:LOGic[:PODA]:BUS:ASSignment

Function Sets or queries the logic input bus assignment.

Syntax :LOGic[:PODA]:BUS:
ASSignment {<String>}
:LOGic[:PODA]:BUS:ASSignment?
<String> = Up to 32 characters

Example :LOGIC:PODA:BUS:ASSIGNMENT "BIT1"
:LOGIC:PODA:BUS:ASSIGNMENT?
-> :LOGIC:PODA:BUS:ASSIGNMENT "BIT1"

:LOGic[:PODA]:BUS:DISPlay

Function Sets or queries the on/off state of the logic input bus display.

Syntax :LOGic[:PODA]:BUS:
DISPlay {<Boolean>}
:LOGic[:PODA]:BUS:DISPlay?

Example :LOGIC:PODA:BUS:DISPLAY ON
:LOGIC:PODA:BUS:DISPLAY?
-> :LOGIC:PODA:BUS:DISPLAY ON

:LOGic[:PODA]:BUS:FORMat

Function Sets or queries the logic input bus display format.

Syntax :LOGic[:PODA]:BUS:FORMat {BINary|
HEX}
:LOGic[:PODA]:BUS:FORMat?

Example :LOGIC:PODA:BUS:FORMAT HEX
:LOGIC:PODA:BUS:FORMAT?
-> :LOGIC:PODA:BUS:FORMAT HEX

:LOGic[:PODA]:BUS:LABel

Function Sets or queries the label of the logic input bus label.

Syntax :LOGic[:PODA]:BUS:LABel {<String>}
:LOGic[:PODA]:BUS:LABel?
<String> = Up to 8 characters

Example :LOGIC:PODA:BUS:LABEL "LOGIC"
:LOGIC:PODA:BUS:LABEL?
-> :LOGIC:PODA:BUS:LABEL "LOGIC"

:LOGic[:PODA]:DESKew

Function Sets or queries the deskewing of the specified pod of the logic input.

Syntax :LOGic[:PODA]:DESKew {<Time>}
:LOGic[:PODA]:DESKew?
<Time> = -1000 ns to 1000 ns (10 ps steps)

Example :LOGIC:PODA:DESKEW 1NS
:LOGIC:PODA:DESKEW?
-> :LOGIC:PODA:DESKEW 1.000E-09

:LOGic[:PODA]:HYSTeresis

Function Sets or queries the hysteresis of the specified pod of the logic input.

Syntax :LOGic[:PODA]:HYSTeresis {HIGH|LOW}
:LOGic[:PODA]:HYSTeresis?

Example :LOGIC:PODA:HYSTERESIS HIGH
:LOGIC:PODA:HYSTERESIS?
-> :LOGIC:PODA:HYSTERESIS HIGH

:LOGic[:PODA]:STATe?

Function Queries all settings related to the logic input state display.

Syntax :LOGic[:PODA]:STATe?

Example :LOGIC:PODA:STATE?

:LOGic[:PODA]:STATe:ASSignment?

Function Queries the logic input state display assignment.

Syntax :LOGic[:PODA]:STATe:ASSignment?

Example :LOGIC:PODA:STATE:ASSIGNMENT?

:LOGic[:PODA]:STATe:ASSignment:ALL

Function Sets or queries all settings related to the logic input state display assignment.

Syntax :LOGic[:PODA]:STATe:ASSignment:
ALL {<Boolean>}
:LOGic[:PODA]:STATe:ASSignment:ALL?

Example :LOGIC:PODA:STATE:ASSIGNMENT:ALL ON
:LOGIC:PODA:STATE:ASSIGNMENT:ALL?
-> :LOGIC:PODA:STATE:ASSIGNMENT:
ALL 1

:LOGic[:PODA]:STATe:ASSignment:BIT<x>

Function Sets or queries the logic input state display assignments of all bits.

Syntax :LOGic[:PODA]:STATe:ASSignment:
BIT<x> {<Boolean>}
:LOGic[:PODA]:STATe:ASSignment:
BIT<x>?
<x> = 0 to 7

Example :LOGIC:PODA:STATE:ASSIGNMENT:BIT1 ON
:LOGIC:PODA:STATE:ASSIGNMENT:BIT1?
-> :LOGIC:PODA:STATE:ASSIGNMENT:
BIT1 1

:LOGic[:PODA]:STATe:ASSignment:BUS

Function Sets or queries the bus setting of the logic input state display assignment.

Syntax :LOGic[:PODA]:STATe:ASSignment:
BUS {<Boolean>}
:LOGic[:PODA]:STATe:ASSignment:BUS?

Example :LOGIC:PODA:STATE:ASSIGNMENT:BUS ON
:LOGIC:PODA:STATE:ASSIGNMENT:BUS?
-> :LOGIC:PODA:STATE:ASSIGNMENT:
BUS 1

:LOGic[:PODA]:STATe:CLOCK

Function Sets or queries the state display reference clock waveform of the logic input.

Syntax :LOGic[:PODA]:STATe:CLOCK {<NRF>|
BIT<x>}
:LOGic[:PODA]:STATe:CLOCK?
<NRF> = 1 to 3
<x> = 1 to 8

Example :LOGIC:PODA:STATE:CLOCK 1
:LOGIC:PODA:STATE:CLOCK?
-> :LOGIC:PODA:STATE:CLOCK 1

:LOGic[:PODA]:STATe:HYSTeresis

Function Sets or queries the hysteresis of the state display reference clock waveform of the logic input.

Syntax :LOGic[:PODA]:STATe:
HYSTeresis {<NRF>}
:LOGic[:PODA]:STATe:HYSTeresis?
<NRF> = 0.3 to 8.0(div)

Example :LOGIC:PODA:STATE:HYSTERESIS 1.0
:LOGIC:PODA:STATE:HYSTERESIS?
-> :LOGIC:PODA:STATE:
HYSTERESIS 1.000E+00

:LOGic[:PODA]:STATe:MODE

Function Sets or queries the on/off state of the logic input state display.

Syntax :LOGic[:PODA]:STATe:
MODE {<Boolean>}
:LOGic[:PODA]:STATe:MODE?

Example :LOGIC:PODA:STATE:MODE ON
:LOGIC:PODA:STATE:MODE?
-> :LOGIC:PODA:STATE:MODE 1

5.18 LOGic group

:LOGic[:PODA]:STAtE:POLarity

Function Sets or queries the polarity of the state display reference clock waveform of the logic input.

Syntax :LOGic[:PODA]:STAtE:POLarity {RISE|FALL|BOTH}
:LOGic[:PODA]:STAtE:POLarity?

Example :LOGIC:PODA:STATE:POLARITY RISE
:LOGIC:PODA:STATE:POLARITY?
-> :LOGIC:PODA:STATE:POLARITY RISE

:LOGic[:PODA]:STAtE:THReshold

Function Sets or queries the detection level of the state display reference clock waveform of the logic input.

Syntax :LOGic[:PODA]:STAtE:THReshold {<NRf>}
:LOGic[:PODA]:STAtE:THReshold?
<NRf> = See the Features Guide for this information.

Example :LOGIC:PODA:STATE:THRESHOLD 1
:LOGIC:PODA:STATE:THRESHOLD?
-> :LOGIC:PODA:STATE:THRESHOLD 1.000E+00

:LOGic:POSition

Function Sets or queries the vertical position of the logic signal.

Syntax :LOGic:POSition {<NRf>}
:LOGic:POSition?
<NRf> = -27 to 80

Example :LOGIC:POSITION 0
:LOGIC:POSITION?
-> :LOGIC:POSITION 0

:LOGic:SIZE

Function Sets or queries the display size of the logic signal.

Syntax :LOGic:SIZE {LARGE|MEDIUM|SMALL}
:LOGic:SIZE?

Example :LOGIC:SIZE LARGE
:LOGIC:SIZE?
-> :LOGIC:SIZE LARGE

5.19 MATH Group

User-defined computation is an option. User-defined computation commands cannot be used on models without the user-defined computation option.

:MATH<x>?

Function Queries all computation settings.

Syntax :MATH<x>?
<x> = 1 to 4

:MATH<x>:DISPlay

Function Sets or queries whether to show (ON) or hide (OFF) the computed waveforms.

Syntax :MATH<x>:DISPlay {<Boolean>}
:MATH<x>:DISPlay?
<x> = 1 to 4

Example :MATH1:DISPLAY ON
:MATH1:DISPLAY?
-> :MATH1:DISPLAY 1

:MATH<x>:ECOUNT?

Function Queries all edge count settings.

Syntax :MATH<x>:ECOUNT?
<x> = 1 to 4

:MATH<x>:ECOUNT:HYSTERESIS

Function Sets or queries the edge detection level hysteresis for edge count.

Syntax :MATH<x>:ECOUNT:HYSTERESIS {<Nrf>}
:MATH<x>:ECOUNT:HYSTERESIS?
<x> = 1 to 4
<Nrf> = 0.3 to 8.0 (div)

Example :MATH1:ECOUNT:HYSTERESIS 1
:MATH1:ECOUNT:HYSTERESIS?
-> :MATH1:ECOUNT:
HYSTERESIS 1.000E+00

:MATH<x>:ECOUNT:POLarity

Function Sets or queries the edge detection polarity for edge count.

Syntax :MATH<x>:ECOUNT:POLarity {FALL|RISE}
:MATH<x>:ECOUNT:POLarity?
<x> = 1 to 4

Example :MATH1:ECOUNT:POLARITY FALL
:MATH1:ECOUNT:POLARITY?
-> :MATH1:ECOUNT:POLARITY FALL

:MATH<x>:ECOUNT:THReshold

Function Sets or queries the edge detection Level for edge count computation.

Syntax :MATH<x>:ECOUNT:THReshold {<Nrf>|
<Voltage>|<Current>}
:MATH<x>:ECOUNT:THReshold?
<x> = 1 to 4

<Voltage>, <Current> = See the Features Guide for this information.

Example :MATH1:ECOUNT:THRESHOLD 1
:MATH1:ECOUNT:THRESHOLD?
-> :MATH1:ECOUNT:THRESHOLD 1.000E+00

:MATH<x>:FILTer?

Function Queries all filter settings.

Syntax :MATH<x>:FILTer?
<x> = 1 to 4

:MATH<x>:FILTer:FORDER

Function Sets or queries the IIR filter's filter order.

Syntax :MATH<x>:FILTer:FORDER {<Nrf>}
:MATH<x>:FILTer:FORDER?
<x> = 1 to 4

<Nrf> = 1 to 2

Example :MATH1:FILTER:FORDER 1
:MATH1:FILTER:FORDER?
-> :MATH1:FILTER:FORDER 1

:MATH<x>:FILTer:HCUTOff

Function Sets or queries the cutoff frequency of the IIR filter high-pass filter.

Syntax :MATH<x>:FILTer:HCUTOff {<Frequency>}
:MATH<x>:FILTer:HCUTOff?
<x> = 1 to 4

<Frequency> = See the Features Guide for this information.

Example :MATH1:FILTER:HCUTOFF 10MHZ
:MATH1:FILTER:HCUTOFF?
-> :MATH1:FILTER:HCUTOFF 10.00E+06

5.19 MATH Group

:MATH<x>:FILTER:LCUToff

Function Sets or queries the cutoff frequency of the IIR filter low-pass filter.

Syntax :MATH<x>:FILTER:LCUToff {<Frequency>}
:MATH<x>:FILTER:LCUToff?
<x> = 1 to 4
<Frequency> = See the Features Guide for this information.

Example :MATH1:FILTER:LCUTOFF 10HZ
:MATH1:FILTER:LCUTOFF?
-> :MATH1:FILTER:LCUTOFF 10.00E+00

:MATH<x>:FILTER:TIME

Function Sets or queries the phase shift delay.

Syntax :MATH<x>:FILTER:TIME {<Time>}
:MATH<x>:FILTER:TIME?
<x> = 1 to 4
<Time> = See the Features Guide for this information.

Example :MATH1:FILTER:TIME 1S
:MATH1:FILTER:TIME?
-> :MATH1:FILTER:TIME 1.000E+00

:MATH<x>:FILTER:TYPE

Function Sets or queries the filter type.

Syntax :MATH<x>:FILTER:TYPE {DElay|MAVG|IHPass|ILPass}
:MATH<x>:FILTER:TYPE?
<x> = 1 to 4

Example :MATH1:FILTER:TYPE DELAY
:MATH1:FILTER:TYPE?
-> :MATH1:FILTER:TYPE DELAY

:MATH<x>:FILTER:WEIGHT

Function Sets or queries the moving average weight.

Syntax :MATH<x>:FILTER:WEIGHT {<NRF>}
:MATH<x>:FILTER:WEIGHT?
<x> = 1 to 4
<NRF> = 2 to 128, 2ⁿ steps

Example :MATH1:FILTER:WEIGHT 2
:MATH1:FILTER:WEIGHT?
-> :MATH1:FILTER:WEIGHT 2

:MATH<x>:INTEgral?

Function Queries all integral settings.

Syntax :MATH<x>:INTEgral?
<x> = 1 to 4

:MATH<x>:INTEgral:SPOint

Function Sets or queries the integral start position.

Syntax :MATH<x>:INTEgral:SPOint {<NRF>}
:MATH<x>:INTEgral:SPOint?
<x> = 1 to 4
<NRF> = -5 to 5 (div)

Example :MATH1:INTEGRAL:SPOINT 1.5
:MATH1:INTEGRAL:SPOINT?
-> :MATH1:INTEGRAL:SPOINT 1.5000000

:MATH<x>:LABel?

Function Queries all settings related to the label of the computed waveform.

Syntax :MATH<x>:LABel?
<x> = 1 to 4

:MATH<x>:LABel[:DEFine]

Function Sets or queries a computed waveform label.

Syntax :MATH<x>:LABel[:DEFine] {<String>}
:MATH<x>:LABel[:DEFine]?
<x> = 1 to 4
<String> = Up to 8 characters

Example :MATH1:LABEL:DEFINE "MATH1"
:MATH1:LABEL:DEFINE?
-> :MATH1:LABEL:DEFINE "MATH1"

:MATH<x>:LABel:MODE

Function Sets or queries the label display on/off state of computed waveforms.

Syntax :MATH<x>:LABel:MODE {<Boolean>}
:MATH<x>:LABel:MODE?
<x> = 1 to 4

Example :MATH1:LABEL:MODE ON
:MATH1:LABEL:MODE?
-> :MATH1:LABEL:MODE 1

:MATH<x>:OPERation

Function Sets or queries the operator.

Syntax :MATH<x>:OPERation {ECOUNT|FILTER|INTEGRal|MINus|MULTiple|PLUS|RCOUNT|USERdefine},{<NRf>|MATH<y>}[, {<NRf>|MATH<y>}][, {<NRf>|MATH<y>}|NONE]
:MATH<x>:OPERation?
<x> = 1 to 4
<NRf> = 1 to 4
<y> = 1 to 4

Example :MATH1:OPERATION PLUS,1,2
:MATH1:OPERATION?
-> :MATH1:OPERATION PLUS,1,2

Description • For unary operators (ECOUNT|FILTER|INTEGRal), select the source waveform with the first <NRf>.

- For binary operators (MINus|MULTiple|PLUS), select the source waveform for the first and second term with the first and second <NRf>, respectively.
- For ternary operators (RCOUNT), select the source waveform for the first, second, and third terms with the first, second, and third <NRf>, respectively.
- For the USERdefine operator, <NRf> is not necessary.

:MATH<x>:RCOUNT?

Function Queries all rotary count operation settings.

Syntax :MATH<x>:RCOUNT?
<x> = 1 to 4

:MATH<x>:RCOUNT:COUNT?

Function Queries all count condition settings for the specified channel's rotary count operation.

Syntax :MATH<x>:RCOUNT:COUNT?
<x> = 1 to 4

:MATH<x>:RCOUNT:COUNT:CWTiming<y>

Function Sets or queries the clockwise (CW) count timing for rotary count operation.

Syntax :MATH<x>:RCOUNT:COUNT:
CWTiming<y> {ARISE|AFALL|BRISe|BFALL}
:MATH<x>:RCOUNT:COUNT:CWTiming<y>?
<x> = 1 to 4
<y> = 1 to 2

Example :MATH1:RCOUNT:COUNT:CWTIMING1 ARISE
:MATH1:RCOUNT:COUNT:CWTIMING1?
-> :MATH1:RCOUNT:COUNT:
CWTIMING1 ARISE

:MATH<x>:RCOUNT:COUNT:MULTi

Function Sets or queries the multiplication for rotary count operation.

Syntax :MATH<x>:RCOUNT:COUNT:MULTi <NRf>
:MATH<x>:RCOUNT:COUNT:MULTi?
<x> = 1 to 4
<NRf> = 1, 2, 4

Example :MATH1:RCOUNT:COUNT:MULTI 4
:MATH1:RCOUNT:COUNT:MULTI?
-> :MATH1:RCOUNT:COUNT:MULTI 4

:MATH<x>:RCOUNT:HYSTeresis<y>

Function Sets or queries the hysteresis for rotary count operation.

Syntax :MATH<x>:RCOUNT:
HYSTeresis<y> {<NRf>}
:MATH<x>:RCOUNT:HYSTeresis<y>?
<x> = 1 to 4
<y> = 1 to 3
<NRf> = 0.3 to 8.0 (div)

Example :MATH1:RCOUNT:HYSTERESIS1 0.3
:MATH1:RCOUNT:HYSTERESIS1?
-> :MATH1:RCOUNT:HYSTERESIS1 0.3000

:MATH<x>:RCOUNT:RESet?

Function Queries all reset condition settings for the specified channel's rotary count operation.

Syntax :MATH<x>:RCOUNT:RESet?
<x> = 1 to 4

:MATH<x>:RCOUNT:RESet:CWTiming

Function Sets or queries the clockwise (CW) reset timing for rotary count operation.

Syntax :MATH<x>:RCOUNT:RESet:
CWTiming {ARISE|AFALL|BRISe|BFALL}
:MATH<x>:RCOUNT:RESet:CWTiming?
<x> = 1 to 4

Example :MATH1:RCOUNT:RESET:CWTIMING ARISE
:MATH1:RCOUNT:RESET:CWTIMING?
-> :MATH1:RCOUNT:RESET:
CWTIMING ARISE

:MATH<x>:RCOUNT:RESet:RTiming

Function Sets or queries the reset timing for rotary count operation.

Syntax :MATH<x>:RCOUNT:RESet:
RTiming {ZONLY|EONZ}
:MATH<x>:RCOUNT:RESet:RTiming?
<x> = 1 to 4

Example :MATH1:RCOUNT:RESET:RTIMING ZONLY
:MATH1:RCOUNT:RESET:RTIMING?
-> :MATH1:RCOUNT:RESET:RTIMING ZONLY

5.19 MATH Group

:MATH<x>:RCOUNT:THReshold<y>

Function Sets or queries the determination level for rotary count operation.

Syntax :MATH<x>:RCOUNT:THReshold<y> {<Nrf>|<Voltage>|<Current>}
:MATH<x>:RCOUNT:THReshold<y>?
<x> = 1 to 4
<y> = 1 to 3
<Voltage>, <Current> = See the Features Guide for this information.

Example :MATH1:RCOUNT:THRESHOLD1 1
:MATH1:RCOUNT:THRESHOLD1?
-> :MATH1:RCOUNT:
THRESHOLD1 1.000E+00

:MATH<x>:RCOUNT:ZPOLarity

Function Sets or queries the polarity for rotary count operation.

Syntax :MATH<x>:RCOUNT:ZPOLarity {NEGative|Positive}
:MATH<x>:RCOUNT:ZPOLarity?
<x> = 1 to 4

Example :MATH1:RCOUNT:ZPOLARITY NEGATIVE
:MATH1:RCOUNT:ZPOLARITY?
-> :MATH1:RCOUNT:ZPOLARITY NEGATIVE

:MATH<x>:SCALE?

Function Queries all scaling settings.

Syntax :MATH<x>:SCALE?
<x> = 1 to 4

:MATH<x>:SCALE:CENTer

Function Sets or queries the center value for manual scaling.

Syntax :MATH<x>:SCALE:CENTer {<Nrf>}
:MATH<x>:SCALE:CENTer?
<x> = 1 to 4
<Nrf> = -1.0000E+31 to 1.0000E+31

Example :MATH1:SCALE:CENTER 1
:MATH1:SCALE:CENTER?
-> :MATH1:SCALE:CENTER 1.00000E+00

:MATH<x>:SCALE:MODE

Function Sets or queries the scale mode.

Syntax :MATH<x>:SCALE:MODE {AUTO|MANual}
:MATH<x>:SCALE:MODE?
<x> = 1 to 4

Example :MATH1:SCALE:MODE AUTO
:MATH1:SCALE:MODE?
-> :MATH1:SCALE:MODE AUTO

:MATH<x>:SCALE:SENSitivity

Function Sets or queries the sensitivity for manual scaling.

Syntax :MATH<x>:SCALE:SENSitivity {<Nrf>}
:MATH<x>:SCALE:SENSitivity?
<x> = 1 to 4
<Nrf> = 1.0000E-31 to 1.0000E+31

Example :MATH1:SCALE:SENSITIVITY 10
:MATH1:SCALE:SENSITIVITY?
-> :MATH1:SCALE:
SENSITIVITY 10.0000E+00

:MATH<x>:UNIT?

Function Queries all computation unit settings.

Syntax :MATH<x>:UNIT?
<x> = 1 to 4

:MATH<x>:UNIT[:DEFine]

Function Sets or queries the computation unit.

Syntax :MATH<x>:UNIT[:DEFine] {<String>}
:MATH<x>:UNIT[:DEFine]?
<x> = 1 to 4
<String> = Up to 4 characters

Example :MATH1:UNIT:DEFINE "EU"
:MATH1:UNIT:DEFINE?
-> :MATH1:UNIT:DEFINE "EU"

Description The unit is applied to scaled values. This command has no effect on the computation result.

:MATH<x>:UNIT:MODE

Function Sets or queries whether a computation unit will be attached automatically or manually.

Syntax :MATH<x>:UNIT:MODE {AUTO|USERdefine}
:MATH<x>:UNIT:MODE?
<x> = 1 to 4

Example :MATH1:UNIT:MODE AUTO
:MATH1:UNIT:MODE?
-> :MATH1:UNIT:MODE AUTO

:MATH<x>:USERdefine?

Function Queries all user-defined computation settings.

Syntax :MATH<x>:USERdefine?
<x> = 1 to 4

:MATH<x>:USERdefine:AVERage?

Function Queries all settings related to the averaging of user-defined computation.

Syntax :MATH<x>:USERdefine:AVERage?
<x> = 1 to 4

Description This command applies to MATH1 to MATH4.

:MATH<x>:USERdefine:AVERage:EWEight

Function Sets or queries the attenuation constant of exponential averaging of user-defined computation.

Syntax :MATH<x>:USERdefine:AVERage:EWEight {<Nrf>}
:MATH<x>:USERdefine:AVERage:EWEight?
<x> = 1 to 4
<Nrf> = 2 to 1024, 2ⁿ steps

Example :MATH1:USERDEFINE:AVERAGE:EWEIGHT 2
:MATH1:USERDEFINE:AVERAGE:EWEIGHT?
-> :MATH1:USERDEFINE:AVERAGE:EWEIGHT 2

Description This command applies to MATH1 to MATH4.

:MATH<x>:USERdefine:AVERage:MODE

Function Sets or queries the averaging mode for user-defined computation.

Syntax :MATH<x>:USERdefine:AVERage:MODE {EXPonent|OFF}
:MATH<x>:USERdefine:AVERage:MODE?
<x> = 1 to 4

Example :MATH1:USERDEFINE:AVERAGE:MODE EXPONENT
:MATH1:USERDEFINE:AVERAGE:MODE?
-> :MATH1:USERDEFINE:AVERAGE:MODE EXPONENT

Description This command applies to MATH1 to MATH4.

:MATH<x>:USERdefine:CONSitant<y>

Function Sets or queries a constant for user-defined computation.

Syntax :MATH<x>:USERdefine:CONSitant<y> {<Nrf>}
:MATH<x>:USERdefine:CONSitant<y>?
<x> = 1 to 4
<y> = 1 to 4
<Nrf> = -1.0000E+31 to 1.0000E+31

Example :MATH1:USERDEFINE:CONSITANT1 1
:MATH1:USERDEFINE:CONSITANT1?
-> :MATH1:USERDEFINE:CONSITANT1 1.0000E+00

Description This command applies to MATH1 to MATH4.

:MATH<x>:USERdefine:DEFine

Function Sets or queries an expression for user-defined computation.

Syntax :MATH<x>:USERdefine:DEFine {<String>}
:MATH<x>:USERdefine:DEFine?
<x> = 1 to 4
<String> = Up to 128 characters

Example :MATH1:USERDEFINE:DEFINE "C1-C2"
:MATH1:USERDEFINE:DEFINE?
-> :MATH1:USERDEFINE:DEFINE "C1-C2"

Description You can only use the characters and symbols on the keyboard that appears on the instrument's screen.

:MATH<x>:USERdefine:FILTer<y>?

Function Queries all settings related to the filter of user-defined computation.

Syntax :MATH<x>:USERdefine:FILTer<y>?
<x> = 1 to 4
<y> = 1 to 2

Description This command applies to MATH1 to MATH4.

:MATH<x>:USERdefine:FILTer<y>:BAND

Function Sets or queries the filter band (frequency band) for user-defined computation.

Syntax :MATH<x>:USERdefine:FILTer<y>:BAND {BPASS|HPASS|LPASS}
:MATH<x>:USERdefine:FILTer<y>:BAND?
<x> = 1 to 4
<y> = 1 to 2

Example :MATH1:USERDEFINE:FILTER1:BAND BPASS
:MATH1:USERDEFINE:FILTER1:BAND?
-> :MATH1:USERDEFINE:FILTER1:BAND BPASS

Description This command applies to MATH1 to MATH4.

:MATH<x>:USERdefine:FILTer<y>:CUTOff<z>

Function Sets or queries an cutoff frequency for user-defined computation.

Syntax :MATH<x>:USERdefine:FILTer<y>:CUTOff<z> {<Nrf>}
:MATH<x>:USERdefine:FILTer<y>:CUTOff<z>?
<x> = 1 to 4
<y> = 1 to 2
<z> = 1 to 2
<Nrf> = 2 to 30(%), 0.2 steps

Example :MATH1:USERDEFINE:FILTER1:CUTOFF1 10
:MATH1:USERDEFINE:FILTER1:CUTOFF1?
-> :MATH1:USERDEFINE:FILTER1:CUTOFF1 10.0

Description CUTOff2 can only be specified with :MATH<x>:USERdefine:FILTer<x>:BAND BPASS. This command applies to MATH1 to MATH4.

5.19 MATH Group

:MATH<x>:USERdefine:FILTer<y>:TYPE

Function Sets or queries the filter type for user-defined computation.

Syntax :MATH<x>:USERdefine:FILTer<y>:
TYPE {IIR|FIR}
:MATH<x>:USERdefine:FILTer<y>:TYPE?
<x> = 1 to 4
<y> = 1 to 2

Example :MATH1:USERDEFINE:FILTER1:TYPE IIR
:MATH1:USERDEFINE:FILTER1:TYPE?
-> :MATH1:USERDEFINE:FILTER1:
TYPE IIR

Description This command applies to MATH1 to MATH4.

:MATH<x>:USERdefine:HISTory:ABORt

Function Aborts the user-defined computation of history waveforms (Math on History).

Syntax :MATH<x>:USERdefine:HISTory:ABORt
<x> = 1 to 4

Example :MATH1:USERDEFINE:HISTORY:ABORT

:MATH<x>:USERdefine:HISTory:EXECute

Function Executes the user-defined computation of history waveforms (Math on History).

Syntax :MATH<x>:USERdefine:HISTory:EXECute
<x> = 1 to 4

Example :MATH1:USERDEFINE:HISTORY:EXECUTE

:MATH<x>:USERdefine:SCALE?

Function Queries all settings related to the scaling of user-defined computation.

Syntax :MATH<x>:USERdefine:SCALE?
<x> = 1 to 4

:MATH<x>:USERdefine:SCALE:ARANging

Function Executes auto range on user-defined computation.

Syntax :MATH<x>:USERdefine:SCALE:ARANging
<x> = 1 to 4

Example :MATH1:USERDEFINE:SCALE:ARANGING

:MATH<x>:USERdefine:SCALE:CENTer

Function Sets or queries the center value of scaling for user-defined computation.

Syntax :MATH<x>:USERdefine:SCALE:
CENTer {<Nrf>}
:MATH<x>:USERdefine:SCALE:CENTer?
<x> = 1 to 4
<Nrf> = -1.0000E+31 to 1.0000E+31

Example :MATH1:USERDEFINE:SCALE:CENTER 1
:MATH1:USERDEFINE:SCALE:CENTER?
-> :MATH1:USERDEFINE:SCALE:
CENTER 1.00000E+00

:MATH<x>:USERdefine:SCALE:SENSitivity

Function Sets or queries the scaling sensitivity for user-defined computation.

Syntax :MATH<x>:USERdefine:SCALE:
SENSitivity {<Nrf>}
:MATH<x>:USERdefine:SCALE:
SENSitivity?
<x> = 1 to 4
<Nrf> = -1.0000E+31 to 1.0000E+31

Example :MATH1:USERDEFINE:SCALE:
SENSITIVITY 10
:MATH1:USERDEFINE:SCALE:SENSITIVITY?
-> :MATH1:USERDEFINE:SCALE:
SENSITIVITY 10.0000E+00

5.20 MEASure Group

The commands dealing with logic waveforms in this group cannot be used on models without logic input.

:MEASure?

Function Queries all the settings for automated measurement of waveform parameters.
Syntax :MEASure?

:MEASure:BIT<x>?

Function Queries all the parameter on/off states of the specified logic waveform.
Syntax :MEASure:BIT<x>?
<x> = 1 to 8

:MEASure:BIT<x>:ALL

Function Collectively turns on or off all measurement items of a logic waveform.
Syntax :MEASure:BIT<x>:ALL {<Boolean>}
<x> = 1 to 8
Example :MEASURE:BIT1:ALL ON

:MEASure:BIT<x>:AREA2?

Function Queries all the parameter on/off states of a logic waveform in Area2.
Syntax :MEASure:BIT<x>:AREA2?
<x> = 1 to 8

:MEASure:BIT<x>:AREA2:ALL

Function Collectively turns on or off all measurement items of a logic waveform in Area2.
Syntax :MEASure:BIT<x>:AREA2:ALL {<Boolean>}
<x> = 1 to 8
Example :MEASURE:BIT1:AREA2:ALL ON

:MEASure:BIT<x>:AREA2:<Parameter>?

Function Queries a waveform parameter setting of a logic waveform in Area2.
Syntax :MEASure:BIT<x>:AREA2:<Parameter>?
<x> = 1 to 8
<Parameter> = {AVGFreq|DElay|DUTYcycle|FREQuency|PERiod|PNUmber}

:MEASure:BIT<x>:AREA2:<Parameter>:COUNT?

Function Queries the statistical processing count of a logic waveform parameter in Area2.
Syntax :MEASure:BIT<x>:AREA2:<Parameter>:COUNT?
<x> = 1 to 8
<Parameter> = {AVGFreq|DElay|DUTYcycle|FREQuency|PERiod|PNUmber}
Example The following is an example of the average frequency of bit 1.
:MEASURE:BIT1:AREA2:AVGFREQ:COUNT?
-> :MEASURE:BIT1:AREA2:AVGFREQ:COUNT 100

:MEASure:BIT<x>:AREA2:<Parameter>:{MAXimum|MEAN|MINimum|SDEviation}?

Function Queries a statistical value of a logic waveform parameter in Area2.
Syntax :MEASure:BIT<x>:AREA2:<Parameter>:{MAXimum|MEAN|MINimum|SDEviation}?
<x> = 1 to 8
<Parameter> = {AVGFreq|DElay|DUTYcycle|FREQuency|PERiod|PNUmber}
Example The following is an example of the average frequency of bit 1.
:MEASURE:BIT1:AREA2:AVGFREQ:MAXIMUM?
-> :MEASURE:BIT1:AREA2:AVGFREQ:MAXIMUM 10.0000E+03

Description If the statistical value is immeasurable, the instrument returns "NAN" (not a number).

:MEASure:BIT<x>:AREA2:<Parameter>:STATe

Function Sets or queries the on/of state of a logic waveform parameter in Area2.
Syntax :MEASure:BIT<x>:AREA2:<Parameter>:STATe {<Boolean>}
:MEASure:BIT<x>:AREA2:<Parameter>:STATe?
<x> = 1 to 8
<Parameter> = {AVGFreq|DUTYcycle|FREQuency|PERiod|PNUmber}
Example The following is an example of the average frequency of bit 1.
:MEASURE:BIT1:AREA2:AVGFREQ:STATE ON
:MEASURE:BIT1:AREA2:AVGFREQ:STATE?
-> :MEASURE:BIT1:AREA2:AVGFREQ:STATE 1

5.20 MEASure Group

:MEASure:BIT<x>:AREA2:<Parameter>:VALUE?

Function Queries automated measurement values of waveform parameters of a logic waveform in Area2.

Syntax :MEASure:BIT<x>:AREA2:<Parameter>:VALUE? [{<NRf>}]
 <x> = 1 to 8
 <Parameter> = {AVGFreq|DElay|DUTYcycle|FREQuency|PERiod|PNUmber}
 <NRf> = See the Features Guide for this information.

Example The following is an example of the average frequency of bit 1.

```
:MEASURE:BIT1:AREA2:AVGFREQ:VALUE?
-> :MEASURE:BIT1:AREA2:AVGFREQ:
VALUE 10.0000E+03
```

Description

- If the value is immeasurable, the instrument returns "NAN" (not a number).
- The <NRf> is used to specify which iteration of automated measurement to query the measured value from.
- If <NRf> is set to 1, the oldest measured value in the automated measurement memory is queried.
- If a measured value does not exist at the specified iteration, the instrument returns "NAN" (not a number).
- If <NRf> is omitted, the most recent measured value is queried.
- If cyclic statistical processing for automated measurement is being executed and <NRf> is specified, the measured values over a cycle in iteration <NRf> from the left of the display on the displayed waveform is queried. If <NRf> is omitted, the measured values over the last cycle on the displayed waveform is queried.

:MEASure:BIT<x>:AREA2:COPY

Function Copies the on/off state of all measurement items of a logic waveform to all other waveforms in Area2.

Syntax :MEASure:BIT<x>:AREA2:COPY
 <x> = 1 to 8

Example :MEASURE:BIT1:AREA2:COPY

:MEASure:BIT<x>:AREA2:DElay?

Function Queries all settings related to the delay between channels of a logic waveform in Area2.

Syntax :MEASure:BIT<x>:AREA2:DElay?
 <x> = 1 to 8

:MEASure:BIT<x>:AREA2:DElay:MEASure?

Function Queries all of the source waveform settings for measuring the delay between channels of a logic waveform in Area2.

Syntax :MEASure:BIT<x>:AREA2:DElay:MEASure?
 <x> = 1 to 8

:MEASure:BIT<x>:AREA2:DElay:MEASure:COUNT

Function Sets or queries the count number of a source waveform for measuring the delay between channels of a logic waveform in Area2.

Syntax :MEASure:BIT<x>:AREA2:DElay:MEASure:COUNT {<NRf>}
 :MEASure:BIT<x>:AREA2:DElay:MEASure:COUNT?
 <x> = 1 to 8
 <NRf> = 1 to 99

Example :MEASURE:BIT1:AREA2:DElay:MEASURE:COUNT 2
 :MEASURE:BIT1:AREA2:DElay:MEASURE:COUNT?
 -> :MEASURE:BIT1:AREA2:DElay:MEASURE:COUNT 2

:MEASure:BIT<x>:AREA2:DElay:MEASure:SLOPe

Function Sets or queries the slope of a source waveform for measuring the delay between channels of a logic waveform in Area2.

Syntax :MEASure:BIT<x>:AREA2:DElay:MEASure:SLOPe {FALL|RISE}
 :MEASure:BIT<x>:AREA2:DElay:MEASure:SLOPe?
 <x> = 1 to 8

Example :MEASURE:BIT1:AREA2:DElay:MEASURE:SLOPE FALL
 :MEASURE:BIT1:AREA2:DElay:MEASURE:SLOPE?
 -> :MEASURE:BIT1:AREA2:DElay:MEASURE:SLOPE FALL

:MEASure:BIT<x>:AREA2:DElay:REfERENCE?

Function Queries all of the reference waveform settings for measuring the delay between channels of a logic waveform in Area2.

Syntax :MEASure:BIT<x>:AREA2:DElay:REFERENCE?
 <x> = 1 to 8

:MEASure:BIT<x>:AREA2:DElay:REFErence:COUNT

Function Sets or queries the count number of the reference waveform for measuring the delay between channels of a logic waveform in Area2.

Syntax :MEASure:BIT<x>:AREA2:DElay:
REFErence:COUNT {<NRF>}
:MEASure:BIT<x>:AREA2:DElay:
REFErence:COUNT?
<x> = 1 to 8
<NRF> = 1 to 99

Example :MEASURE:BIT1:AREA2:DELAY:REFERENCE:
COUNT 2
:MEASURE:BIT1:AREA2:DELAY:REFERENCE:
COUNT?
-> :MEASURE:BIT1:AREA2:DELAY:
REFERENCE:COUNT 2

:MEASure:BIT<x>:AREA2:DElay:REFErence:SLOPe

Function Sets or queries the slope of the reference waveform for measuring the delay between channels of a logic waveform in Area2.

Syntax :MEASure:BIT<x>:AREA2:DElay:
REFErence:SLOPe {FALL|RISE}
:MEASure:BIT<x>:AREA2:DElay:
REFErence:SLOPe?
<x> = 1 to 8

Example :MEASURE:BIT1:AREA2:DELAY:REFERENCE:
SLOPE FALL
:MEASURE:BIT1:AREA2:DELAY:REFERENCE:
SLOPE?
-> :MEASURE:BIT1:AREA2:DELAY:
REFERENCE:SLOPE FALL

:MEASure:BIT<x>:AREA2:DElay:REFErence:SOURce

Function Sets or queries whether the reference point for measuring the delay between channels of a logic waveform will be set to the trigger point or a waveform in Area2.

Syntax :MEASure:BIT<x>:AREA2:DElay:
REFErence:SOURce {TRACE|TRIGger}
:MEASure:BIT<x>:AREA2:DElay:
REFErence:SOURce?
<x> = 1 to 8

Example :MEASURE:BIT1:AREA2:DELAY:REFERENCE:
SOURCE TRACE
:MEASURE:BIT1:AREA2:DELAY:REFERENCE:
SOURCE?
-> :MEASURE:BIT1:AREA2:DELAY:
REFERENCE:SOURCE TRACE

:MEASure:BIT<x>:AREA2:DElay:REFErence:TRACe

Function Sets or queries the edge of the reference waveform for measuring the delay between channels of a logic waveform in Area2.

Syntax :MEASure:BIT<x>:AREA2:DElay:
REFErence:TRACe {<NRF>|BIT<x>|
MATH<x>}
:MEASure:BIT<x>:AREA2:DElay:
REFErence:TRACe?
BIT<x>'s <x> = 1 to 8
<NRF> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:BIT1:AREA2:DELAY:REFERENCE:
TRACE 2
:MEASURE:BIT1:AREA2:DELAY:REFERENCE:
TRACE?
-> :MEASURE:BIT1:AREA2:DELAY:
REFERENCE:TRACE 2

:MEASure:BIT<x>:AREA2:DElay:STATe

Function Sets or queries the on/off state and display format of the delay parameter in Area2.

Syntax :MEASure:BIT<x>:AREA2:DElay:
STATe {OFF|ON|DEGRee}
:MEASure:BIT<x>:AREA2:DElay:STATe?
<x> = 1 to 8

Example :MEASURE:BIT1:AREA2:DELAY:STATE ON
:MEASURE:BIT1:AREA2:DELAY:STATE?
-> :MEASURE:BIT1:AREA2:DELAY:
STATE ON

:MEASure:BIT<x>:<Parameter>?

Function Queries a waveform parameter setting of a logic waveform.

Syntax :MEASure:BIT<x>:<Parameter>?
<x> = 1 to 8
<Parameter> = {AVGFreq|DElay|DUTYcycle|FREQuency|PERiod|PNUMber}

:MEASure:BIT<x>:<Parameter>:COUNT?

Function Queries the statistical processing count of a logic waveform parameter.

Syntax :MEASure:BIT<x>:<Parameter>:COUNT?
<x> = 1 to 8
<Parameter> = {AVGFreq|DElay|DUTYcycle|FREQuency|PERiod|PNUMber}

Example The following is an example of the average frequency of bit 1.
:MEASURE:BIT1:AVGFREQ:COUNT?
-> :MEASURE:BIT1:AVGFREQ:COUNT 100

5.20 MEASure Group

:MEASure:BIT<x>:<Parameter>:{MAXimum|MEAN|MINimum|SDEVIation}?

Function	Queries a statistical value of a logic waveform parameter.
Syntax	<pre>:MEASure:BIT<x>:<Parameter>:{MAXimum MEAN MINimum SDEVIation}? <x> = 1 to 8 <Parameter> = {AVGFreq DElay DUTYcycle FREQuency PERiod PNUmber}</pre>
Example	The following is an example of the average frequency of bit 1. <pre>:MEASURE:BIT1:AVGFREQ:MAXIMUM? -> :MEASURE:BIT1:AVGFREQ: MAXIMUM 10.0000E+03</pre>
Description	If the statistical value cannot be obtained, the instrument returns "NAN" (not a number).

:MEASure:BIT<x>:<Parameter>:STATe

Function	Sets or queries the on/of state of a logic waveform parameter.
Syntax	<pre>:MEASure:BIT<x>:<Parameter>: STATe {<Boolean>} :MEASure:BIT<x>:<Parameter>:STATe? <x> = 1 to 8 <Parameter> = {AVGFreq DUTYcycle FREQuency PERiod PNUmber}</pre>
Example	The following is an example of the average frequency of bit 1. <pre>:MEASURE:BIT1:AVGFREQ:STATE ON :MEASURE:BIT1:AVGFREQ:STATE? -> :MEASURE:BIT1:AVGFREQ:STATE 1</pre>

:MEASure:BIT<x>:<Parameter>:VALue?

Function	Queries automated measurement values of waveform parameters of a logic waveform.
Syntax	<pre>:MEASure:BIT<x>:<Parameter>: VALue? [{<NRf>}] <x> = 1 to 8 <Parameter> = {AVGFreq DElay DUTYcycle FREQuency PERiod PNUmber} <NRf> = See the Features Guide for this information.</pre>
Example	The following is an example of the average frequency of bit 1. <pre>:MEASURE:BIT1:AVGFREQ:VALUE? -> :MEASURE:BIT1:AVGFREQ: VALUE 10.0000E+03</pre>

Description If the value is immeasurable, the instrument returns "NAN" (not a number). The <NRf> at the end is used to specify which iteration of statistical processing to query the parameter value from. This command returns "NAN" if the specified value does not exist.

- For measurements other than cyclic statistical processing <NRf> can be omitted. If you omit, the most recent history parameter value will be queried. If <NRf> is included, the parameter value of the waveform <NRf> times before the newest waveform will be queried from the acquisition memory.
- For measurements after cyclic statistical processing <NRf> can be omitted. If you omit, the parameter value within the cycle that was measured last will be queried. If you include <NRf>, the parameter value within the cycle that was measured <NRf> times after the measurement shown on the screen's left edge will be queried.

:MEASure:BIT<x>:COPY

Function	Copies the on/off state of all measurement items of a logic waveform to all other waveforms.
Syntax	<pre>:MEASure:BIT<x>:COPY <x> = 1 to 8</pre>
Example	<pre>:MEASURE:BIT1:COPY</pre>

:MEASure:BIT<x>:DElay?

Function	Queries all settings related to the delay between channels of a logic waveform.
Syntax	<pre>:MEASure:BIT<x>:DElay? <x> = 1 to 8</pre>

:MEASure:BIT<x>:DElay:MEASure?

Function	Queries all the settings for a source waveform for measuring the delay between channels of a logic waveform.
Syntax	<pre>:MEASure:BIT<x>:DElay:MEASure? <x> = 1 to 8</pre>

:MEASure:BIT<x>:DELay:MEASure:COUNT

Function Sets or queries the source waveform count for measuring the delay between channels of a logic waveform.

Syntax :MEASure:BIT<x>:DELay:MEASure:
COUNT {<NRf>}
:MEASure:BIT<x>:DELay:MEASure:COUNT?
<x> = 1 to 8
<NRf> = 1 to 99

Example :MEASURE:BIT1:DELAY:MEASURE:COUNT 2
:MEASURE:BIT1:DELAY:MEASURE:COUNT?
-> :MEASURE:BIT1:DELAY:MEASURE:
COUNT 2

:MEASure:BIT<x>:DELay:MEASure:SLOPe

Function Sets or queries a source waveform slope that will be used to measure delay between channels for a logic waveform.

Syntax :MEASure:BIT<x>:DELay:MEASure:
SLOPe {FALL|RISE}
:MEASure:BIT<x>:DELay:MEASure:SLOPe?
<x> = 1 to 8

Example :MEASURE:BIT1:DELAY:MEASURE:
SLOPE FALL
:MEASURE:BIT1:DELAY:MEASURE:SLOPE?
-> :MEASURE:BIT1:DELAY:MEASURE:
SLOPE FALL

:MEASure:BIT<x>:DELay:REFeRence?

Function Queries all the settings for the reference waveform for measuring the delay between channels of a logic waveform.

Syntax :MEASure:BIT<x>:DELay:REFeRence?
<x> = 1 to 8

:MEASure:BIT<x>:DELay:REFeRence:COUNT

Function Sets or queries the reference waveform count for measuring the delay between channels of a logic waveform.

Syntax :MEASure:BIT<x>:DELay:REFeRence:
COUNT {<NRf>}
:MEASure:BIT<x>:DELay:REFeRence:
COUNT?
<x> = 1 to 8
<NRf> = 1 to 99

Example :MEASURE:BIT1:DELAY:REFERENCE:
COUNT 2
:MEASURE:BIT1:DELAY:REFERENCE:COUNT?
-> :MEASURE:BIT1:DELAY:REFERENCE:
COUNT 2

:MEASure:BIT<x>:DELay:REFeRence:SLOPe

Function Sets or queries the reference waveform slope that will be used to measure delay between channels for a logic waveform.

Syntax :MEASure:BIT<x>:DELay:REFeRence:
SLOPe {FALL|RISE}
:MEASure:BIT<x>:DELay:REFeRence:
SLOPe?
<x> = 1 to 8

Example :MEASURE:BIT1:DELAY:REFERENCE:
SLOPE FALL
:MEASURE:BIT1:DELAY:REFERENCE:SLOPE?
-> :MEASURE:BIT1:DELAY:REFERENCE:
SLOPE FALL

:MEASure:BIT<x>:DELay:REFeRence:SOURce

Function Sets or queries whether the reference point for measuring the delay between channels of a logic waveform will be set to the trigger point or a logic waveform.

Syntax :MEASure:BIT<x>:DELay:REFeRence:
SOURce {TRACe|TRIGger}
:MEASure:BIT<x>:DELay:REFeRence:
SOURce?
<x> = 1 to 8

Example :MEASURE:BIT1:DELAY:REFERENCE:
SOURCE TRACE
:MEASURE:BIT1:DELAY:REFERENCE:
SOURCE?
-> :MEASURE:BIT1:DELAY:REFERENCE:
SOURCE TRACE

:MEASure:BIT<x>:DELay:REFeRence:TRACe

Function Sets or queries the reference waveform edge that will be used to measure delay between channels for a logic waveform.

Syntax :MEASure:BIT<x>:DELay:REFeRence:
TRACe {<NRf>|BIT<y>|MATH<y>}
:MEASure:BIT<x>:DELay:REFeRence:
TRACe?
<x> = 1 to 8
<NRf> = 1 to 4
<y> of BIT<y> = 1 to 8
<y> of MATH<y> = 1 to 4

Example :MEASURE:BIT1:DELAY:REFERENCE:
TRACE 2
:MEASURE:BIT1:DELAY:REFERENCE:TRACE?
-> :MEASURE:BIT1:DELAY:REFERENCE:
TRACE 2

5.20 MEASure Group

:MEASure:BIT<x>:DELAY:STATE

Function Sets or queries the on/off state and display format of the delay parameter.

Syntax :MEASure:BIT<x>:DELAY:STATE {OFF|ON|DEGREE}
:MEASure:BIT<x>:DELAY:STATE?
<x> = 1 to 8

Example :MEASURE:BIT1:DELAY:STATE ON
:MEASURE:BIT1:DELAY:STATE?
-> :MEASURE:BIT1:DELAY:STATE ON

:MEASure:{CHANnel<x>|MATH<x>}:?

Function Queries all the parameter on/off states of a waveform.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

:MEASure:{CHANnel<x>|MATH<x>}:ALL

Function Collectively turns on or off all measurement items of a waveform.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:
ALL {<Boolean>}
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:ALL ON

:MEASure:{CHANnel<x>|MATH<x>}:AREA2?

Function Queries all the parameter on/off states of a waveform in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:ALL

Function Collectively turns on or off all measurement items of a waveform in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:
ALL {<Boolean>}
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:AREA2:ALL ON

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:<Parameter>?

Function Queries a waveform parameter setting of a waveform in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:
<Parameter>?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4
<Parameter> = {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth|DELay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDth|PERiod|PNUMBER|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEVIation|TY1Integ|TY2Integ|V1|V2}

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:<Parameter>:COUNT?

Function Queries the statistical processing count of a waveform parameter in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:
<Parameter>:COUNT?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4
<Parameter> = {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth|DELay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDth|PERiod|PNUMBER|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEVIation|TY1Integ|TY2Integ|V1|V2}

Example The following is an example of the average value of CH1.

```
:MEASURE:CHANNEL1:AREA2:AVERAGE:
COUNT?
-> :MEASURE:CHANNEL1:AREA2:AVERAGE:
COUNT 100
```

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:<Parameter>:{MAXimum|MEAN|MINimum|SDEVIation}?

Function Queries a statistical value of a waveform parameter in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:<Parameter>:{MAXimum|MEAN|MINimum|SDEVIation}?

CHANnel<x>'s <x> = 1 to 4

MATH<x>'s <x> = 1 to 4

<Parameter> = {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth|DELay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDth|PERiod|PNUMber|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEVIation|TY1Integ|TY2Integ|V1|V2}

Example The following is an example of the average value of CH1.

```
:MEASURE:CHANNEL1:AREA2:AVERAGE:
MAXIMUM?
-> :MEASURE:CHANNEL1:AREA2:AVERAGE:
MAXIMUM 10.0000E+03
```

Description If the statistical value cannot be obtained, the instrument returns "NAN" (not a number).

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:<Parameter>:STATe

Function Sets or queries the on/of state of a waveform parameter in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:<Parameter>:STATe {<Boolean>}

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:<Parameter>:STATe?

CHANnel<x>'s <x> = 1 to 4

MATH<x>'s <x> = 1 to 4

<Parameter> = {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth|DELay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDth|PERiod|PNUMber|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEVIation|TY1Integ|TY2Integ|V1|V2}

Example The following is an example of the average value of CH1.

```
:MEASURE:CHANNEL1:AREA2:AVERAGE:
STATE ON
:MEASURE:CHANNEL1:AREA2:AVERAGE:
STATE?
-> :MEASURE:CHANNEL1:AREA2:AVERAGE:
STATE 1
```

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:<Parameter>:VALue?

Function Queries the automated measurement value of a waveform parameter in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:<Parameter>:VALue? [<NRF>[,STATUS]]

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:<Parameter>:VALue? [STATUS]

CHANnel<x>'s <x> = 1 to 4

MATH<x>'s <x> = 1 to 4

<Parameter> = {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth|DELay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWIDth|PERiod|PNUMber|POVershoot|PTOPeak|PWIDth|RISE|RMS|SDEVIation|TY1Integ|TY2Integ|V1|V2}

<NRF> = See the Features Guide for this information.

Example The following is an example of the average value of CH1.

```
:MEASURE:CHANNEL1:AREA2:AVERAGE:
VALUE?
-> :MEASURE:CHANNEL1:AREA2:AVERAGE:
VALUE 10.0000E+03
```

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:CO PY

Function Copies the on/off state of all measurement items of a waveform to all other waveforms in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:CO PY

CHANnel<x>'s <x> = 1 to 4

MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:AREA2:CO PY

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:CY CLe

Function Sets or queries cycle mode of Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:CY CLe {ONE|N|OFF}

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:CY CLe?

CHANnel<x>'s <x> = 1 to 4

MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:AREA2:CY CLe ONE

:MEASURE:CHANNEL1:AREA2:CY CLe?

```
-> :MEASURE:CHANNEL1:AREA2:CY CLe ONE
```

Description • When :ANALysis:PANalyze:TYPE SWLoss is specified, "SWLoss" is returned in response to a query.

- If :ANALysis:PMEasure<x>:MODE ON or :ANALysis:PMEasure:CYCMode NCYCLe is specified, "NCCHannel" is returned in response to a query.

5.20 MEASure Group

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:DE Lay?

Function Queries all settings related to the delay between channels of a waveform in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:DE Lay:MEASure?

Function Queries all of the source waveform settings for measuring the delay between channels of a waveform in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:MEASure?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:DE Lay:MEASure:COUNT

Function Sets or queries the count number of a source waveform for measuring the delay between channels of a waveform in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:MEASure:COUNT {<Nrf>}
:MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:MEASure:COUNT?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4
<Nrf> = 1 to 99

Example :MEASURE:CHANNEL1:AREA2:DELAY:
MEASURE:COUNT 2
:MEASURE:CHANNEL1:AREA2:DELAY:
MEASURE:COUNT?
-> :MEASURE:CHANNEL1:AREA2:DELAY:
MEASURE:COUNT 2

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:DE Lay:MEASure:SLOPe

Function Sets or queries the slope of a source waveform for measuring the delay between channels of a waveform in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:MEASure:SLOPe {FALL|RISE}
:MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:MEASure:SLOPe?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:AREA2:DELAY:
MEASURE:SLOPE FALL
:MEASURE:CHANNEL1:AREA2:DELAY:
MEASURE:SLOPE?
-> :MEASURE:CHANNEL1:AREA2:DELAY:
MEASURE:SLOPE FALL

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:DE Lay:REFeRence?

Function Queries all of the reference waveform settings for measuring the delay between channels of a waveform in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:REFeRence?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:DE Lay:REFeRence:COUNT

Function Sets or queries the count number of the reference waveform for measuring the delay between channels of a waveform in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:REFeRence:COUNT {<Nrf>}
:MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:REFeRence:COUNT?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4
<Nrf> = 1 to 99

Example :MEASURE:CHANNEL1:AREA2:DELAY:
REFERENCE:COUNT 2
:MEASURE:CHANNEL1:AREA2:DELAY:
REFERENCE:COUNT?
-> :MEASURE:CHANNEL1:AREA2:DELAY:
REFERENCE:COUNT 2

:MEASure:{CHANnel<x>|MATH<x>}:AREA2:DE Lay:REFeRence:SLOPe

Function Sets or queries the slope of the reference waveform for measuring the delay between channels of a waveform in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:REFeRence:SLOPe {FALL|RISE}
:MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:REFeRence:SLOPe?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:AREA2:DELAY:
REFERENCE:SLOPE FALL
:MEASURE:CHANNEL1:AREA2:DELAY:
REFERENCE:SLOPE?
-> :MEASURE:CHANNEL1:AREA2:DELAY:
REFERENCE:SLOPE FALL

**:MEASure:{CHANnel<x>|MATH<x>}:AREA2:DE
Lay:REfERENCE:SOURce**

Function Sets or queries whether the reference point for measuring the delay between channels of a waveform will be set to the trigger point or a waveform in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:REfERENCE:SOURce {TRACe|
TRIGger}
:MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:REfERENCE:SOURce?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:AREA2:DELAY:
REFERENCE:SOURCE TRACE
:MEASURE:CHANNEL1:AREA2:DELAY:
REFERENCE:SOURCE?
-> :MEASURE:CHANNEL1:AREA2:DELAY:
REFERENCE:SOURCE TRACE

**:MEASure:{CHANnel<x>|MATH<x>}:AREA2:DE
Lay:REfERENCE:TRACe**

Function Sets or queries the edge of the reference waveform for measuring the delay between channels of a waveform in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:REfERENCE:TRACe {<NRf>|
MATH<x>}
:MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:REfERENCE:TRACe?
<NRf> = 1 to 4
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:AREA2:DELAY:
REFERENCE:TRACE 2
:MEASURE:CHANNEL1:AREA2:DELAY:
REFERENCE:TRACE?
-> :MEASURE:CHANNEL1:AREA2:DELAY:
REFERENCE:TRACE 2

**:MEASure:{CHANnel<x>|MATH<x>}:AREA2:DE
Lay:STATe**

Function Sets or queries the on/off state and display format of the delay parameter in Area2.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:STATe {OFF|ON|DEGRee}
:MEASure:{CHANnel<x>|MATH<x>}:AREA2:
DElay:STATe?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:AREA2:DELAY:
STATE ON
:MEASURE:CHANNEL1:AREA2:DELAY:STATE?
-> :MEASURE:CHANNEL1:AREA2:DELAY:
STATE ON

**:MEASure:{CHANnel<x>|MATH<x>}:<Paramet
er>?**

Function Queries a waveform parameter setting of a waveform.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:
<Parameter>?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4
<Parameter> = {AMPLitude|AVERAge|AVGFreq
|AVGPeriod|BWIDTH|DElay|DT|DUTYcycle|ENU
Mber|FALL|FREQuency|HIGH|LOW|MAXimum|M
INimum|NOVershoot|NWIDTH|PERiod|PNUMber
|POVershoot|PTOPeak|PWIDTH|RISE|RMS|SDEV
iation|TY1Integ|TY2Integ|V1|V2}

**:MEASure:{CHANnel<x>|MATH<x>}:<Paramet
er>:COUNT?**

Function Queries the statistical processing count of a waveform parameter.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:
<Parameter>:COUNT?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4
<Parameter> = {AMPLitude|AVERAge|AVGFreq
|AVGPeriod|BWIDTH|DElay|DT|DUTYcycle|ENU
Mber|FALL|FREQuency|HIGH|LOW|MAXimum|M
INimum|NOVershoot|NWIDTH|PERiod|PNUMber
|POVershoot|PTOPeak|PWIDTH|RISE|RMS|SDEV
iation|TY1Integ|TY2Integ|V1|V2}

Example The following is an example of the average value of CH1.
:MEASURE:CHANNEL1:AVERAGE:COUNT?
-> :MEASURE:CHANNEL1:AVERAGE:
COUNT 100

**:MEASure:{CHANnel<x>|MATH<x>}:<Paramet
er>:{MAXimum|MEAN|MINimum|SDEViation)?**

Function Queries a statistical value of a waveform parameter.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:
<Parameter>:{MAXimum|MEAN|MINimum|
SDEViation)?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4
<Parameter> = {AMPLitude|AVERAge|AVGFreq
|AVGPeriod|BWIDTH|DElay|DT|DUTYcycle|ENU
Mber|FALL|FREQuency|HIGH|LOW|MAXimum|M
INimum|NOVershoot|NWIDTH|PERiod|PNUMber
|POVershoot|PTOPeak|PWIDTH|RISE|RMS|SDEV
iation|TY1Integ|TY2Integ|V1|V2}

Example The following is an example of the average value of CH1.
:MEASURE:CHANNEL1:AVERAGE:MAXIMUM?
-> :MEASURE:CHANNEL1:AVERAGE:
MAXIMUM 10.0000E+03

Description If the statistical value cannot be obtained, the instrument returns "NAN" (not a number).

5.20 MEASure Group

:MEASure:{CHANnel<x>|MATH<x>}:<Parameter>:STATe

Function	Sets or queries the on/of state of a waveform parameter.
Syntax	<pre>:MEASure:{CHANnel<x> MATH<x>}: <Parameter>:STATe {<Boolean>} :MEASure:{CHANnel<x> MATH<x>}: <Parameter>:STATe? CHANnel<x>'s <x> = 1 to 4 MATH<x>'s <x> = 1 to 4 <Parameter> = {AMPLitude AVERAge AVGFreq AVGPeriod BWIDth DELay DT DUTYcycle ENU Mber FALL FREQuency HIGH LOW MAXimum M INimum NOVershoot NWIDTH PERiod PNUMber POVershoot PTOPeak PWIDTH RISE RMS SDEV iation TY1Integ TY2Integ V1 V2}</pre>
Example	The following is an example of the average value of CH1. <pre>:MEASURE:CHANNEL1:AVERAGE:STATE ON :MEASURE:CHANNEL1:AVERAGE:STATE? -> :MEASURE:CHANNEL1:AVERAGE:STATE 1</pre>

:MEASure:{CHANnel<x>|MATH<x>}:<Parameter>:VALue?

Function	Queries automated measurement values of waveform parameters of a waveform.
Syntax	<pre>:MEASure:{CHANnel<x> MATH<x>}: <Parameter>:VALue? [<NRf>[,STATus]] :MEASure:{CHANnel<x> MATH<x>}: <Parameter>:VALue? [STATus] CHANnel<x>'s <x> = 1 to 4 MATH<x>'s <x> = 1 to 4 <Parameter> = {AMPLitude AVERAge AVGFreq AVGPeriod BWIDth DELay DT DUTYcycle ENU Mber FALL FREQuency HIGH LOW MAXimum M INimum NOVershoot NWIDTH PERiod PNUMber POVershoot PTOPeak PWIDTH RISE RMS SDEV iation TY1Integ TY2Integ V1 V2} <NRf> = See the Features Guide for this information.</pre>
Example	The following is an example of the average value of CH1. <pre>:MEASURE:CHANNEL1:AVERAGE:VALUE? -> :MEASURE:CHANNEL1:AVERAGE: VALUE 10.0000E+03</pre>
Description	• If the value is immeasurable, the instrument returns "NAN" (not a number). • The <NRf> is used to specify which iteration of automated measurement to query the measured value from. • If <NRf> is set to 1, the oldest measured value in the automated measurement memory is queried. • If a measured value does not exist at the specified iteration, the instrument returns "NAN" (not a number). • If <NRf> is omitted, the most recent measured value is queried. • If cyclic statistical processing for automated measurement is being executed and <NRf> is specified, the measured values over a cycle in iteration <NRf> from the left of the display on the displayed waveform is queried. If <NRf> is omitted, the measured values over the last cycle on the displayed waveform is queried. • The optional "STATus" at the end is valid when the parameter is set to FALL, NWIDTH, PERiod, PWIDTH, or RISE. If "STATus" is omitted for these parameters, if the measurement value is less than the measurement resolution, a negative value is returned. If "STATus" is included, "LOW_RESOL" is returned.
	:MEASure:{CHANnel<x> MATH<x>}:COPY
Function	Copies the on/off state of all measurement items of a waveform to all other waveforms.
Syntax	<pre>:MEASure:{CHANnel<x> MATH<x>}:COPY CHANnel<x>'s <x> = 1 to 4 MATH<x>'s <x> = 1 to 4</pre>
Example	<pre>:MEASURE:CHANNEL1:COPY</pre>

:MEASure:{CHANnel<x>|MATH<x>}:CYCLE

Function Sets or queries the cycle mode.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:
CYCLE {ONE|N|OFF}
:MEASure:{CHANnel<x>|MATH<x>}:CYCLE?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:CYCLE ONE
:MEASURE:CHANNEL1:CYCLE?
-> :MEASURE:CHANNEL1:CYCLE ONE

Description • When :ANALysis:PANalyze:TYPE SWLoss is specified, "SWLoss" is returned in response to a query.
• If :ANALysis:PMEASure<x>:MODE ON or :ANALysis:PMEASure:CYCMode NCYCLE is specified, "NCCChannel" is returned in response to a query.

:MEASure:{CHANnel<x>|MATH<x>}:DELAY?

Function Queries all settings related to the delay between channels of a waveform.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:DELAY?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

:MEASure:{CHANnel<x>|MATH<x>}:DELAY:MEASure?

Function Queries all the settings for a source waveform for measuring the delay between channels of a waveform.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:DELAY:MEASure?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

:MEASure:{CHANnel<x>|MATH<x>}:DELAY:MEASure:COUNT

Function Sets or queries the number of edges at which delay between channels will be measured for a waveform.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:DELAY:MEASure:COUNT {<NRF>}
:MEASure:{CHANnel<x>|MATH<x>}:DELAY:MEASure:COUNT?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4
<NRF> = 1 to 99

Example :MEASURE:CHANNEL1:DELAY:MEASURE:COUNT 2
:MEASURE:CHANNEL1:DELAY:MEASURE:COUNT?
-> :MEASURE:CHANNEL1:DELAY:MEASURE:COUNT 2

:MEASure:{CHANnel<x>|MATH<x>}:DELAY:MEASure:SLOPe

Function Sets or queries a source waveform slope that will be used to measure delay between channels for a waveform.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:DELAY:MEASure:SLOPe {FALL|RISE}
:MEASure:{CHANnel<x>|MATH<x>}:DELAY:MEASure:SLOPe?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:DELAY:MEASURE:SLOPE FALL
:MEASURE:CHANNEL1:DELAY:MEASURE:SLOPE?
-> :MEASURE:CHANNEL1:DELAY:MEASURE:SLOPE FALL

:MEASure:{CHANnel<x>|MATH<x>}:DELAY:REFERENCE?

Function Queries all reference waveform settings used to measure the delay between channels for the specified waveform.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:DELAY:REFERENCE?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

:MEASure:{CHANnel<x>|MATH<x>}:DELAY:REFERENCE:COUNT

Function Sets or queries the reference waveform count for measuring the delay between channels of a waveform.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:DELAY:REFERENCE:COUNT {<NRF>}
:MEASure:{CHANnel<x>|MATH<x>}:DELAY:REFERENCE:COUNT?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4
<NRF> = 1 to 99

Example :MEASURE:CHANNEL1:DELAY:REFERENCE:COUNT 2
:MEASURE:CHANNEL1:DELAY:REFERENCE:COUNT?
-> :MEASURE:CHANNEL1:DELAY:REFERENCE:COUNT 2

5.20 MEASure Group

:MEASure:{CHANnel<x>|MATH<x>}:DElay:RE Ference:SLOPe

Function Sets or queries the reference waveform slope that will be used to measure delay between channels for a waveform.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:DElay:
REFeRence:SLOPe {FALL|RISE}
:MEASure:{CHANnel<x>|MATH<x>}:DElay:
REFeRence:SLOPe?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:DELAY:REFERENCE:
SLOPE FALL
:MEASURE:CHANNEL1:DELAY:REFERENCE:
SLOPE?
-> :MEASURE:CHANNEL1:DELAY:
REFERENCE:SLOPE FALL

:MEASure:{CHANnel<x>|MATH<x>}:DElay:RE Ference:SOURce

Function Sets or queries whether to set the reference point for measuring the delay between channels for a waveform to a trigger point or to a waveform.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:DElay:
REFeRence:SOURce {TRACe|TRIGger}
:MEASure:{CHANnel<x>|MATH<x>}:DElay:
REFeRence:SOURce?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:DELAY:REFERENCE:
SOURCE TRACE
:MEASURE:CHANNEL1:DELAY:REFERENCE:
SOURCE?
-> :MEASURE:CHANNEL1:DELAY:
REFERENCE:SOURCE TRACE

:MEASure:{CHANnel<x>|MATH<x>}:DElay:RE Ference:TRACe

Function Sets or queries the reference waveform edge that will be used to measure delay between channels for a waveform.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:DElay:
REFeRence:TRACe {<NRf>|MATH<x>}
:MEASure:{CHANnel<x>|MATH<x>}:DElay:
REFeRence:TRACe?
<NRf> = 1 to 4
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:DELAY:REFERENCE:
TRACE 2
:MEASURE:CHANNEL1:DELAY:REFERENCE:
TRACE?
-> :MEASURE:CHANNEL1:DELAY:
REFERENCE:TRACE 2

:MEASure:{CHANnel<x>|MATH<x>}:DElay:ST ATe

Function Sets or queries the on/off state and display format of the delay parameter.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:DElay:
STATe {OFF|ON|DEGRee}
:MEASure:{CHANnel<x>|MATH<x>}:DElay:
STATe?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:DELAY:STATE ON
:MEASURE:CHANNEL1:DELAY:STATE?
-> :MEASURE:CHANNEL1:DELAY:STATE ON

:MEASure:{CHANnel<x>|MATH<x>}:DPRoxim al?

Function Queries all distal, mesial, and proximal settings.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:
DPRoximal?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

:MEASure:{CHANnel<x>|MATH<x>}:DPRoxima l:MODE

Function Sets or queries the distal, mesial, and proximal point mode setting.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:
DPRoximal:MODE {PERCent|UNIT}
:MEASure:{CHANnel<x>|MATH<x>}:
DPRoximal:MODE?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:DPROXIMAL:
MODE PERCENT
:MEASURE:CHANNEL1:DPROXIMAL:MODE?
-> :MEASURE:CHANNEL1:DPROXIMAL:
MODE PERCENT

**:MEASure:{CHANnel<x>|MATH<x>}:DPRoxima
l:PERCent**

Function Sets or queries the distal, mesial, and proximal points as percentages.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:
DPRoximal:PERCent {<NRf>,<NRf>,
<NRf>}
:MEASure:{CHANnel<x>|MATH<x>}:
DPRoximal:PERCent?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4
<NRf> = 0 to 100 (% , in steps of 1)

Example :MEASURE:CHANNEL1:DPROXIMAL:
PERCENT 80,60,40
:MEASURE:CHANNEL1:DPROXIMAL:PERCENT?
-> :MEASURE:CHANNEL1:DPROXIMAL:
PERCENT 40,60,80

**:MEASure:{CHANnel<x>|MATH<x>}:DPRoxima
l:UNIT**

Function Sets or queries the distal, mesial, and proximal points as voltages.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:
DPRoximal:UNIT {<Voltage>,<Voltage>,
<Voltage>|<NRf>,<NRf>,<NRf>}
:MEASure:{CHANnel<x>|MATH<x>}:
DPRoximal:UNIT?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4
<Voltage> , <NRf> = See the Features Guide for
this information.

Example :MEASURE:CHANNEL1:DPROXIMAL:
UNIT 50V,0V,-50V
:MEASURE:CHANNEL1:DPROXIMAL:UNIT?
-> :MEASURE:CHANNEL1:DPROXIMAL:
UNIT -50.0E+00,0.0E+00,50.0E+00

Description When a current probe is in use, this command does not set or query the <Current> value.

:MEASure:{CHANnel<x>|MATH<x>}:METHod

Function Sets or queries the high and low points.

Syntax :MEASure:{CHANnel<x>|MATH<x>}:
METHod {AUTO|MAXMin|HISTogram}
:MEASure:{CHANnel<x>|MATH<x>}:
METHod?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CHANNEL1:METHod AUTO
:MEASURE:CHANNEL1:METHod?
-> :MEASURE:CHANNEL1:METHod AUTO

:MEASure:CONTInuous?

Function Queries all settings for the normal statistical processing of automatically measured waveform parameters.

Syntax :MEASure:CONTInuous?

:MEASure:CONTInuous:REStart

Function Restarts the normal statistical processing of automatically measured waveform parameters.

Syntax :MEASure:CONTInuous:REStart
:MEASURE:CONTINUOUS:RESTART

**:MEASure:CONTInuous:TLCHange
(TriggerLevel Change)**

Function Sets or queries whether normal statistical processing of automatically measured waveform parameters is restarted when the trigger level is changed.

Syntax :MEASure:CONTInuous:
TLCHange{REStart|IGNore}:MEASure:
CONTInuous:TLCHange?

Example :MEASURE:CONTINUOUS:
TLCHANGE RESTART:MEASURE:CONTINUOUS:
TLCHANGE?
-> :MEASURE:CONTINUOUS:
TLCHANGE RESTART

:MEASure:CYCLe?

Function Queries all settings related to the cyclic statistical processing of automated measurement.

Syntax :MEASure:CYCLe?

:MEASure:CYCLe:ABORT

Function Aborts the execution of the cyclic statistical processing of automated measurement.

Syntax :MEASure:CYCLe:ABORT

Example :MEASURE:CYCLE:ABORT

:MEASure:CYCLe:EXECute

Function Executes the cyclic statistical processing of automated measurement.

Syntax :MEASure:CYCLe:EXECute

Example :MEASURE:CYCLE:EXECUTE

:MEASure:CYCLe:TRACe

Function Sets or queries the source waveform of the cyclic statistical processing of automated measurement.

Syntax :MEASure:CYCLe:TRACe {OWN|<NRf>|
MATH<x>}
:MEASure:CYCLe:TRACe?
<NRf> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :MEASURE:CYCLE:TRACE 1
:MEASURE:CYCLE:TRACE?
-> :MEASURE:CYCLE:TRACE 1

5.20 MEASure Group

:MEASure:HISTory:ABORT

Function Aborts the execution of the statistical processing of the history waveform of automated measurement.

Syntax :MEASure:HISTory:ABORT
:MEASURE:HISTORY:ABORT

:MEASure:HISTory:EXECute

Function Executes the statistical processing of the history waveform of automated measurement.

Syntax :MEASure:HISTory:EXECute
Example :MEASURE:HISTORY:EXECUTE

:MEASure:INDicator

Function Sets or queries the measurement location indicator.

Syntax :MEASure:INDicator {<NRF>|BIT<x>|
MATH<x>|OFF{,<Parameter>,2}}
:MEASure:INDicator?
<NRF> = 1 to 4
BIT<x>'s <x> = 1 to 8
MATH<x>'s <x> = 1 to 4
<Parameter> = {AMPLitude|AVERage|AVGFreq|
AVGPeriod|BWidth|DELay|DUTYcycle|FALL|FR
EQuency|HIGH|LOW|MAXimum|MINimum|NOV
ershoot|NWidth|PERiod|POVershoot|PTOPeak
|PWidth|RISE|RMS|V1|V2}

Example :MEASURE:INDICATOR 1,AVERAGE
:MEASURE:INDICATOR?
-> :MEASURE:INDICATOR 1,AVERAGE

:MEASure:MODE

Function Sets or queries the automated measurement mode.

Syntax :MEASure:MODE {OFF|ON|CONTinuous|
CYCLE|HISTory}
:MEASure:MODE?

Example :MEASURE:MODE OFF
:MEASURE:MODE?
-> :MEASURE:MODE OFF

:MEASure:RANGe<x>

Function Sets or queries the measurement source window.

Syntax :MEASure:RANGe<x> {MAIN|Z1|Z2}
:MEASure:RANGe<x>?
<x> = 1 to 2

Example :MEASURE:RANGE1 MAIN
:MEASURE:RANGE1?
-> :MEASURE:RANGE1 MAIN

Description ":Measure:RANGe1" is for setting or querying the measurement source window of Area1 for normal measurement and two area measurement. ":Measure:RANGe2" is for setting or querying the measurement source window of Area2 for two area measurement.

:MEASure:TRANGe<x> (Time Range)

Function Sets or queries the measurement time period.

Syntax :MEASure:TRANGe<x> {<NRF>,<NRF>}
:MEASure:TRANGe<x>?
<x> = 1 to 2
<NRF> = -5 to 5 (div), 10 div/display record length steps

Example :MEASURE:TRANGE1 -4,4
:MEASURE:TRANGE1?
-> :MEASURE:TRANGE1 -4.00E+00,
4.00E+00

Description ":Measure:TRANGe1" is for setting or querying the measurement range of Area1 for normal measurement and two area measurement. ":Measure:TRANGe2" is for setting or querying the measurement range of Area2 for two area measurement.

:MEASure:USER<x>?

Function Queries all automatic measurement settings for a Calc item.

Syntax :MEASure:USER<x>?
<x> = 1 to 4

:MEASure:USER<x>:COUNT?

Function Queries the statistical processing count of the automatically measured value of a Calc item.

Syntax :MEASure:USER<x>:COUNT?
<x> = 1 to 4

Example :MEASURE:USER1:COUNT?
-> :MEASURE:USER1:COUNT 100

:MEASure:USER<x>:DEFine

Function Sets or queries the expression for the automatically measured value of a Calc item.

Syntax :MEASure:USER<x>:DEFine {<String>}
:MEASure:USER<x>:DEFine?
<x> = 1 to 4
<String> = Up to 128 characters

Example :MEASURE:USER1:DEFINE "MAX(C1)"
:MEASURE:USER1:DEFINE?
-> :MEASURE:USER1:DEFINE "MAX(C1)"

:MEASure:USER<x>:{MAXimum|MEAN|MINimum|SDEVIation}?

Function Queries a statistical value that is calculated on the automatically measured value of a Calc item.

Syntax :MEASure:USER<x>:MAXimum?
<x> = 1 to 4

Example The following example is for the maximum value.
:MEASURE:USER1:MAXIMUM?
-> :MEASURE:USER1:MAXIMUM 1.000E+00

Description If the statistical value cannot be obtained, the instrument returns "NAN" (not a number).

:MEASure:USER<X>:NAME

Function Sets or queries the name of a Calc item.

Syntax :MEASure:USER<X>:NAME {<String>}
 :MEASure:USER<X>:NAME?
 <x> = 1 to 4
 <String> = Up to 8 characters

Example :MEASURE:USER1:NAME "MAX"
 :MEASURE:USER1:NAME?
 -> :MEASURE:USER1:NAME "MAX"

:MEASure:USER<x>:STATE

Function Sets or queries the on/off status of automated measurement of a Calc item.

Syntax :MEASure:USER<x>:STATE {<Boolean>}
 :MEASure:USER<x>:STATE?
 <x> = 1 to 4

Example :MEASURE:USER1:STATE ON
 :MEASURE:USER1:STATE?
 -> :MEASURE:USER1:STATE 1

:MEASure:USER<x>:UNIT

Function Sets or queries the unit of a Calc item.

Syntax :MEASure:USER<x>:UNIT {<String>}
 :MEASure:USER<x>:UNIT?
 <x> = 1 to 4
 <String> = Up to 4 characters

Example :MEASURE:USER1:UNIT "V"
 :MEASURE:USER1:UNIT?
 -> :MEASURE:USER1:UNIT "V"

Description Measured values are never affected by the unit.

:MEASure:USER<x>:VALue?

Function Queries the automatically measured value of a Calc item.

Syntax :MEASure:USER<x>:VALue? [<NRf>]
 <x> = 1 to 4
 <NRf> = See the Features Guide for this information.

Example :MEASURE:USER1:VALUE?
 -> :MEASURE:USER1:VALUE 1.000E+00

Description

- If the value is immeasurable, the instrument returns "NAN" (not a number).
- The <NRf> is used to specify which iteration of automated measurement to query the measured value from.
- If <NRf> is set to 1, the oldest measured value in the automated measurement memory is queried.
- If a measured value does not exist at the specified iteration, the instrument returns "NAN" (not a number).
- If <NRf> is omitted, the most recent measured value is queried.

:MEASure:WAIT?

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

Syntax :MEASure:WAIT? {<NRf>}
 <NRf> = 1 to 36000 (timeout value, 100 ms steps)

Example :MEASURE:WAIT? 10
 -> :MEASURE:WAIT 1

Description

- If the execution of automated measurement is completed within the timeout period, 0 is returned. If it is not completed or if automated measurement is not being performed, 1 is returned.
- Even if you set a long timeout 0 is returned as soon as the automated measurement is completed.

:MEASure:ZLINKage

Function Sets or queries the linkage (on/off) between the measurement numbers and zoom positions of the results of automated measurement of waveform parameters.

Syntax :MEASure:ZLINKage {<Boolean>}
 :MEASure:ZLINKage?

Example :MEASURE:ZLINKAGE ON
 :MEASURE:ZLINKAGE?
 -> :MEASURE:ZLINKAGE 1

5.21 RECall group

:RECall:SETup<x>:EXECute

Function Recalls setup data from the internal memory.

Syntax :RECall:SETup<x>:EXECute

<x> = 1 to 3

Example :RECALL:SETUP1:EXECUTE

Description If you specify an area that does not contain setup data, an error occurs.

5.22 REference Group

:REference<x>?

Function Queries all reference waveform settings.

Syntax :REference<x>?
<x> = 1 to 4

:REference<x>:DISPlay

Function Sets or queries the reference waveform display on/off state.

Syntax :REference<x>:DISPlay {<Boolean>}
:REference<x>:DISPlay?
<x> = 1 to 4

Example :REFERENCE1:DISPLAY ON
:REFERENCE1:DISPLAY?
-> :REFERENCE1:DISPLAY 1

:REference<x>:LABel?

Function Queries all settings related to the label of a reference.

Syntax :REference<x>:LABel
<x> = 1 to 4

:REference<x>:LABel[:DEFine]

Function Sets or queries the label of a reference.

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 Sets or queries the label display on/off state of a reference.

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 a reference waveform.

Syntax :REference<x>:LOAD {<NRf>|MATH<y>}
<x> = 1 to 4
<NRf> = 1 to 4
<y> = 1 to 4

Example :REFERENCE1:LOAD 1

Description • When <X> is 1, <y> can be set between 2 and 4.
• When <X> is 2, <y> can be set 1, 3, or 4.
• When <X> is 3, <y> can be set 1, 2, or 4.
• When <X> is 4, <y> can be set between 1 and 3.

:REference<x>:POSition

Function Sets or queries the vertical position of the reference waveform.

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

5.23 SEARCh Group

The following table indicates how the communication commands for search methods correspond to the search types that appear when you press a panel key.

Search Type	Communication Command	Setup Menu	
		Panel Key	Search Type
Edge	EDGE	SEARCH key	Edge
Multiple input combination (pattern)	PATtern		Pattern
Pulse width	PULSe		Pulse Width
Timeout value	TIMEout		Timeout

:SEARCh?

Function Queries all waveform search settings.

Syntax :SEARCh?

:SEARCh:ABORt

Function Aborts the search.

Syntax :SEARCh:ABORt

Example :SEARCH:ABORT

:SEARCh:ASCROll<x>?

Function Queries all auto scroll settings.

Syntax :SEARCh:ASCROll<x>?

<x> = 1 to 2

:SEARCh:ASCROll<x>:JUMP

Function Moves the center position of the zoom box to the left or right edge of the main window.

Syntax :SEARCh:ASCROll<x>:JUMP {LEFT|RIGHT}

<x> = 1 to 2

Example :SEARCH:ASCROLL1:JUMP LEFT

:SEARCh:ASCROll<x>:SPEEd

Function Sets or queries the auto scroll speed of the zoom box.

Syntax :SEARCh:ASCROll<x>:SPEEd {<NRf>}

:SEARCh:ASCROll<x>:SPEEd?

<x> = 1 to 2

<NRf> = 1 to 50

Example :SEARCH:ASCROLL1:SPEED 1

:SEARCH:ASCROLL1:SPEED?

-> :SEARCH:ASCROLL1:SPEED 1

:SEARCh:ASCROll<x>:STARt

Function Starts auto scrolling.

Syntax :SEARCh:ASCROll<x>:STARt {LEFT|RIGHT}

<x> = 1 to 2

Example :SEARCH:ASCROLL1:START LEFT

:SEARCh:ASCROll<x>:STOP

Function Stops auto scrolling.

Syntax :SEARCh:ASCROll<x>:STOP

<x> = 1 to 2

Example :SEARCH:ASCROLL1:STOP

:SEARCh:EDGE?

Function Queries all edge search settings.

Syntax :SEARCh:EDGE?

:SEARCh:EDGE:HYSTeresis

Function Sets or queries the edge search level hysteresis.

Syntax :SEARCh:EDGE:HYSTeresis {<NRf>}

:SEARCh:EDGE:HYSTeresis?

<NRf> = 0.3 to 8.0 (div), 0.1 div steps

Example :SEARCH:EDGE:HYSTERESIS 1

:SEARCH:EDGE:HYSTERESIS?

-> :SEARCH:EDGE:HYSTERESIS 1.0

:SEARCh:EDGE:LEVel

Function Sets or queries the edge search level.

Syntax :SEARCh:EDGE:LEVel {<NRf>|<Voltage>}

:SEARCh:EDGE:LEVel?

<NRf> = See the Features Guide for this information.

Example :SEARCH:EDGE:LEVEL -10mV

:SEARCH:EDGE:LEVEL?

-> :SEARCH:EDGE:LEVEL -10.00E-03

Description When a current probe is in use, this command does not set or query the <Current> value.

:SEARCh:EDGE:SLOPe

Function Sets or queries the edge search slope.

Syntax :SEARCh:EDGE:SLOPe {BOTH|FALL|RISE}

:SEARCh:EDGE:SLOPe?

Example :SEARCH:EDGE:SLOPE RISE

:SEARCH:EDGE:SLOPE?

-> :SEARCH:EDGE:SLOPE RISE

:SEARCh:EDGE:SOURce

Function Sets or queries the source waveform of edge search.

Syntax :SEARCh:EDGE:SOURce {<NRf>|BIT<x>|
MATH<x>}
:SEARCh:EDGE:SOURce?
<NRf> = 1 to 4
BIT<x>'s <x> = 1 to 8
MATH<x>'s <x> = 1 to 4

Example :SEARCH:EDGE:SOURCE 1
:SEARCH:EDGE:SOURCE?
-> :SEARCH:EDGE:SOURCE 1

:SEARCh:EPOint

Function Sets or queries the search end position.

Syntax :SEARCh:EPOint {<NRf>}
:SEARCh:EPOint?
<NRf> = -5 to 5 (div), 10 div/display record length steps

Example :SEARCH:EPOINT 0
:SEARCH:EPOINT?
-> :SEARCH:EPOINT 0.000E+00

:SEARCh:EXECute

Function Executes a search. This is an overlap command.

Syntax :SEARCh:EXECute
Example :SEARCH:EXECUTE

:SEARCh:MARK

Function Sets or queries the detected point mark on/off state.

Syntax :SEARCh:MARK {<Boolean>}
:SEARCh:MARK?

Example :SEARCH:MARK ON
:SEARCH:MARK?
-> :SEARCH:MARK 1

:SEARCh:MAG<x>

Function Sets or queries the zoom factor of the zoom window.

Syntax :SEARCh:MAG<x> {<NRf>}
:SEARCh:MAG<x>?
<x> = 1 to 2
<NRf> = See the Features Guide for this information.

Example :SEARCH:MAG1 2
:SEARCH:MAG1?
-> :SEARCH:MAG1 2

:SEARCh:POSition<x>

Function Sets or queries the position of a zoom box.

Syntax :SEARCh:POSition<x> {<NRf>}
:SEARCh:POSition<x>?
<x> = 1 to 2
<NRf> = -5 to 5 (div), 10 div/display record length steps

Example :SEARCH:POSITION1 1
:SEARCH:POSITION1?
-> :SEARCH:POSITION1 1.000E+00

:SEARCh:PATtern?

Function Queries all settings related to the combination (pattern) search of multiple inputs.

Syntax :SEARCh:PATtern?

:SEARCh:PATtern:BITS?

Function Queries all settings related to the bits of the combination (pattern) search of multiple inputs.

Syntax :SEARCh:PATtern:BITS?

:SEARCh:PATtern:BITS:HEXa

Function Sets the bit pattern for the combination (pattern) search of multiple inputs in hexadecimal notation.

Syntax :SEARCh:PATtern:BITS:HEXa {<String>}
Example :SEARCH:PATTERN:BITS:HEXA "AB"

:SEARCh:PATtern:BITS:PATtern

Function Sets or queries the bit pattern for the combination (pattern) search of multiple inputs in binary notation.

Syntax :SEARCh:PATtern:BITS:
PATtern {<String>}
:SEARCh:PATtern:BITS:PATtern?

Example :SEARCH:PATTERN:BITS:
PATTERN "1100xx01"
:SEARCH:PATTERN:BITS:PATTERN?
-> :SEARCH:PATTERN:BITS:
PATTERN "1100xx01"

:SEARCh:PATtern:{CHANnel<x>|MATH<x>}?

Function Queries all settings related to a waveform of the combination (pattern) search of multiple inputs.

Syntax :SEARCh:PATtern:{CHANnel<x>|
MATH<x>}?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

5.23 SEARCh Group

:SEARCh:PATtern:{CHANnel<x>|MATH<x>}:HYSTeresis

Function Sets or queries the hysteresis of the source waveform for the combination (pattern) search of multiple inputs.

Syntax :SEARCh:PATtern:{CHANnel<x>|
MATH<x>}:HYSTeresis {<NRF>}
:SEARCh:PATtern:{CHANnel<x>|
MATH<x>}:HYSTeresis?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4
<NRF> = 0.3 to 8.0 (div), 0.1 div steps

Example :SEARCH:PATTERN:CHANNEL1:
HYSTERESIS 0.3
:SEARCH:PATTERN:CHANNEL1:HYSTERESIS?
-> :SEARCH:PATTERN:CHANNEL1:
HYSTERESIS 0.30

:SEARCh:PATtern:{CHANnel<x>|MATH<x>}:LEVel

Function Sets or queries the search level of the source waveform for the combination (pattern) search of multiple inputs.

Syntax :SEARCh:PATtern:{CHANnel<x>|
MATH<x>}:LEVel {<NRF>|<Voltage>}
:SEARCh:PATtern:{CHANnel<x>|
MATH<x>}:LEVel?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4
<NRF> = See the Features Guide for this information.

Example :SEARCH:PATTERN:CHANNEL1:LEVEL 1V
:SEARCH:PATTERN:CHANNEL1:LEVEL?
-> :SEARCH:PATTERN:CHANNEL1:
LEVEL 1.000E+00

Description When a current probe is in use, this command does not set or query the <Current> value.

:SEARCh:PATtern:{CHANnel<x>|MATH<x>}:PATtern

Function Sets or queries the search pattern of the source waveform for the combination (pattern) search of multiple inputs.

Syntax :SEARCh:PATtern:{CHANnel<x>|
MATH<x>}:PATtern {DONTcare|HIGH|LOW}
:SEARCh:PATtern:{CHANnel<x>|
MATH<x>}:PATtern?
CHANnel<x>'s <x> = 1 to 4
MATH<x>'s <x> = 1 to 4

Example :SEARCH:PATTERN:CHANNEL1:
PATTERN HIGH
:SEARCH:PATTERN:CHANNEL1:PATTERN?
-> :SEARCH:PATTERN:CHANNEL1:
PATTERN HIGH

:SEARCh:PATtern:CLOCK?

Function Queries all settings related to the clock channel of the combination (pattern) search of multiple inputs.

Syntax :SEARCh:PATtern:CLOCK?

:SEARCh:PATtern:CLOCK:HYSTeresis

Function Sets or queries the hysteresis of the clock channel for the combination (pattern) search of multiple inputs.

Syntax :SEARCh:PATtern:CLOCK:
HYSTeresis {<NRF>}
:SEARCh:PATtern:CLOCK:HYSTeresis?
<NRF> = 0.3 to 8.0 (div), 0.1 div steps

Example :SEARCH:PATTERN:CLOCK:HYSTERESIS 4.0
:SEARCH:PATTERN:CLOCK:HYSTERESIS?
-> :SEARCH:PATTERN:CLOCK:
HYSTERESIS 4.00

Description When :SEARCh:PATtern:CLOCK:SOURce is set to NONE, an error occurs.

:SEARCh:PATtern:CLOCK:LEVel

Function Sets or queries the search level of the clock channel for the combination (pattern) search of multiple inputs.

Syntax :SEARCh:PATtern:CLOCK:LEVel {<NRF>|
<Voltage>}
:SEARCh:PATtern:CLOCK:LEVel?
<NRF> = See the Features Guide for this information.

Example :SEARCH:PATTERN:CLOCK:LEVEL
:SEARCH:PATTERN:CLOCK:LEVEL?
-> :SEARCH:PATTERN:CLOCK:LEVEL

Description • When :SEARCh:PATtern:CLOCK:SOURce is set to NONE, an error occurs.
• When a current probe is in use, this command does not set or query the <Current> value.

:SEARCh:PATtern:CLOCK:SLOPe

Function Sets or queries the slope of the clock channel for the combination (pattern) search of multiple inputs.

Syntax :SEARCh:PATtern:CLOCK:SLOPe {BOTH|
FALL|RISE}
:SEARCh:PATtern:CLOCK:SLOPe?

Example :SEARCH:PATTERN:CLOCK:SLOPE RISE
:SEARCH:PATTERN:CLOCK:SLOPE?
-> :SEARCH:PATTERN:CLOCK:SLOPE RISE

Description When :SEARCh:PATtern:CLOCK:SOURce is set to NONE, an error occurs.

:SEARCh:PATtern:CLOCk:SOURce

Function Sets or queries the source waveform of the clock channel for the combination (pattern) search of multiple inputs.

Syntax :SEARCh:PATtern:CLOCk:SOURce {<NRf>|
BIT<x>|NONE|MATH<x>}
:SEARCh:PATtern:CLOCk:SOURce?
<NRf> = 1 to 4
BIT<x>'s <x> = 1 to 8
MATH<x>'s <x> = 1 to 4

Example :SEARCH:PATTERN:CLOCK:SOURCE 1
:SEARCH:PATTERN:CLOCK:SOURCE?
-> :SEARCH:PATTERN:CLOCK:SOURCE 1

:SEARCh:PATtern:CONDition

Function Sets or queries the achievement condition of the source waveform for the combination (pattern) search of multiple inputs.

Syntax :SEARCh:PATtern:CONDition {ENTER|
EXIT|FALSE|TRUE}
:SEARCh:PATtern:CONDition?

Example :SEARCH:PATTERN:CONDITION ENTER
:SEARCH:PATTERN:CONDITION?
-> :SEARCH:PATTERN:CONDITION ENTER

:SEARCh:PATtern:LOGic

Function Sets or queries the logic of the source waveform for the combination (pattern) search of multiple inputs.

Syntax :SEARCh:PATtern:LOGic {AND|OR}
:SEARCh:PATtern:LOGic?

Example :SEARCH:PATTERN:LOGIC AND
:SEARCH:PATTERN:LOGIC?
-> :SEARCH:PATTERN:LOGIC AND

:SEARCh:PATtern:TIME<x>

Function Sets or queries the reference time of the source waveform for the combination (pattern) search of multiple inputs.

Syntax :SEARCh:PATtern:TIME<x> {<Time>}
:SEARCh:PATtern:TIME<x>?
<x> = 1 to 2
<Time> = 1 ns to 10 s

Example :SEARCH:PATTERN:TIME1 1S
:SEARCH:PATTERN:TIME1?
-> :SEARCH:PATTERN:TIME1 1.000E+00

:SEARCh:PATtern:TQQualify

Function Sets or queries the time condition of the source waveform for the combination (pattern) of multiple inputs.

Syntax :SEARCh:PATtern:TQQualify {INSide|
LTHan|MTHan|OUTSide|TIMEout}
:SEARCh:PATtern:TQQualify?

Example :SEARCH:PATTERN:TQUALIFY INSIDE
:SEARCH:PATTERN:TQUALIFY?
-> :SEARCH:PATTERN:TQUALIFY INSIDE

:SEARCh:PULSe?

Function Queries all pulse width search settings.

Syntax :SEARCh:PULSe?

:SEARCh:PULSe:HYSTeresis

Function Sets or queries the pulse width search level hysteresis.

Syntax :SEARCh:PULSe:HYSTeresis {<NRf>}
:SEARCh:PULSe:HYSTeresis?
<NRf> = 0.3 to 8.0, 0.1 div steps

Example :SEARCH:PULSE:HYSTERESIS 1
:SEARCH:PULSE:HYSTERESIS?
-> :SEARCH:PULSE:HYSTERESIS 1.0

:SEARCh:PULSe:LEVel

Function Sets or queries the pulse width search level.

Syntax :SEARCh:PULSe:LEVel {<NRf>|<Voltage>}
:SEARCh:PULSe:LEVel?
<NRf> = See the Features Guide for this information.

Example :SEARCH:PULSE:LEVEL 1V
:SEARCH:PULSE:LEVEL?
-> :SEARCH:PULSE:LEVEL 1.000E+00

Description When a current probe is in use, this command does not set or query the <Current> value.

:SEARCh:PULSe:POLarity

Function Sets or queries the pulse width search level polarity.

Syntax :SEARCh:PULSe:POLarity {HIGH|LOW}
:SEARCh:PULSe:POLarity?

Example :SEARCH:PULSE:POLARITY HIGH
:SEARCH:PULSE:POLARITY?
-> :SEARCH:PULSE:POLARITY HIGH

:SEARCh:PULSe:SOURce

Function Sets or queries the source waveform of pulse width search.

Syntax :SEARCh:PULSe:SOURce {<NRf>|BIT<x>|
MATH<x>|NONE}
:SEARCh:PULSe:SOURce?
<NRf> = 1 to 4

BIT<x>'s <x> = 1 to 8
MATH<x>'s <x> = 1 to 4
Example :SEARCH:PULSE:SOURCE 1
:SEARCH:PULSE:SOURCE?
-> :SEARCH:PULSE:SOURCE 1

5.23 SEARch Group

:SEARch:PULSe:TIME<x>

Function Sets or queries the reference time of pulse width search.

Syntax :SEARch:PULSe:TIME<x> {<Time>}
:SEARch:PULSe:TIME<x>?
<x> = 1 to 2
<Time> = 1 ns to 10 s

Example :SEARCH:PULSE:TIME1 1S
:SEARCH:PULSE:TIME1?
-> :SEARCH:PULSE:TIME1 1.000E+00

:SEARch:PULSe:TQQualify

Function Sets or queries the time condition for pulse width searching.

Syntax :SEARch:PULSe:TQQualify {INSide|
LTHan|MTHan|OUTSide|TIMEout}
:SEARch:PULSe:TQQualify?

Example :SEARCH:PULSE:TQUALIFY INSIDE
:SEARCH:PULSE:TQUALIFY?
-> :SEARCH:PULSE:TQUALIFY INSIDE

:SEARch:SELEct

Function Queries the setting of the detected point to show in the zoom window and the zoom position of that point.

Syntax :SEARch:SELEct {<Nrf>|MAXimum}
:SEARch:SELEct?
<Nrf> = 0 to 49999

Example :SEARCH:SELECT 1
:SEARCH:SELECT?
-> :SEARCH:SELECT 1.50000000

Description If there are no detected points, the instrument returns "NAN."

:SEARch:SELEct? MAXimum

Function Queries the total number of detected points.

Syntax :SEARch:SELEct? {MAXimum}
Example :SEARCH:SELECT? MAXIMUM
-> :SEARCH:SELECT 100

Description If there are no detected points, the instrument returns "NAN."

:SEARch:SKIP?

Function Queries all skip mode settings.

Syntax :SEARch:SKIP?

:SEARch:SKIP:DECimation

Function Sets or queries the decimation detection of skip mode.

Syntax :SEARch:SKIP:DECimation {<Nrf>}
:SEARch:SKIP:DECimation?
<Nrf> = 1 to 9999

Example :SEARCH:SKIP:DECIMATION 1
:SEARCH:SKIP:DECIMATION?
-> :SEARCH:SKIP:DECIMATION 1

:SEARch:SKIP:HOLDoff

Function Sets or queries the hold-off detection.

Syntax :SEARch:SKIP:HOLDoff {<Time>}
:SEARch:SKIP:HOLDoff?
<Time> = 0.1 ns to 1 s

Example :SEARCH:SKIP:HOLDOFF 0S
:SEARCH:SKIP:HOLDOFF?
-> :SEARCH:SKIP:HOLDOFF 0.000E+00

:SEARch:SKIP:MODE

Function Sets or queries the mode of skip mode.

Syntax :SEARch:SKIP:MODE {DECimation|
HOLDoff|OFF}
:SEARch:SKIP:MODE?
Example :SEARCH:SKIP:MODE DECIMATION
:SEARCH:SKIP:MODE?
-> :SEARCH:SKIP:MODE DECIMATION

:SEARch:SPOint

Function Sets or queries the search start position.

Syntax :SEARch:SPOint {<Nrf>}
:SEARch:SPOint?
<Nrf> = -5 to 5 (div), 10 div/display record length steps

Example :SEARCH:SPOINT -1
:SEARCH:SPOINT?
-> :SEARCH:SPOINT -1.000E+00

:SEARch:TIMEout?

Function Queries all timeout period search settings.

Syntax :SEARch:TIMEout?

:SEARch:TIMEout:POLarity

Function Sets or queries the timeout period search polarity.

Syntax :SEARch:TIMEout:POLarity {NEGative|
POSitive}
:SEARch:TIMEout:POLarity?

Example :SEARCH:TIMEOUT:POLARITY NRGATIVE
:SEARCH:TIMEOUT:POLARITY?
-> :SEARCH:TIMEOUT:POLARITY NRGATIVE

:SEARCh:TIMEout:SOURce

Function Sets or queries the timeout period search source waveform.

Syntax :SEARCh:TIMEout:SOURce {<NRf>|
MATH<x>|BIT<x>}
:SEARCh:TIMEout:SOURce?
<NRf> = 1 to 4
BIT<x>'s <x> = 1 to 8
MATH<x>'s <x> = 1 to 4

Example :SEARCH:TIMEOUT:SOURCE 1
:SEARCH:TIMEOUT:SOURCE?
-> :SEARCH:TIMEOUT:SOURCE 1

:SEARCh:TIMEout:TIME

Function Sets or queries the timeout period of the timeout period search.

Syntax :SEARCh:TIMEout:TIME <Time>
:SEARCh:TIMEout:TIME?
<Time> = 1 ns to 10 s

Example :SEARCH:TIMEOUT:TIME 1
:SEARCH:TIMEOUT:TIME?
-> :SEARCH:TIMEOUT:TIME 1

:SEARCh:TWINDow

Function Sets or queries the zoom window for showing the detected area.

Syntax :SEARCh:TWINDow {Z1|Z2}
:SEARCh:TWINDow?

Example :SEARCH:TWINDOW Z1
:SEARCH:TWINDOW?
-> :SEARCH:TWINDOW Z1

:SEARCh:TYPE

Function Sets or queries the search type.

Syntax :SEARCh:TYPE {EDGE|PATTeRn|PULSe|
TIMEout}
:SEARCh:TYPE?

Example :SEARCH:TYPE EDGE
:SEARCH:TYPE?
-> :SEARCH:TYPE EDGE

5.24 SERIALbus group

The following table indicates how the communication commands for serial bus types correspond to the serial bus types that appear when you press a panel key.

Serial Bus Type	Communication Command	Setup Menu Panel Key	Serial Bus Type
CAN	CAN*	SHIFT+SEARCH (SERIAL BUS) Key	CAN
CAN FD	CANFD*		CAN FD
CXPI	CXPI*		CXPI
FlexRay	FLEXray*		FlexRay
I2C	I2C*		I2C
LIN	LIN*		LIN
PSI5	PSI5*		PSI5 Airbag
SENT	SENT*		SENT
User-defined	SPATtern		User Define
SPI	SPI*		SPI
UART	UART*		UART

* CAN, CANFD, CXPI, FLEXray, I2C, LIN, PSI5, SENT, SPI, and UART are serial bus options. These serial bus commands cannot be used on models without these options.

:SERIALbus<x>?

Function Queries all serial bus signal analysis and search settings.

Syntax :SERIALbus<x>?
<x> = 1 to 4

:SERIALbus<x>:ASETup?

Function Queries all auto setup settings of serial bus signal analysis.

Syntax :SERIALbus<x>:ASETup?
<x> = 1 to 4

:SERIALbus<x>:ASETup:ABORT

Function Aborts the serial bus signal auto setup.

Syntax :SERIALbus<x>:ASETup:ABORT
<x> = 1 to 4

Example :SERIALBUS1:ASETUP:ABORT

:SERIALbus<x>:ASETup:EXECute

Function Execute an serial bus signal auto setup.

Syntax :SERIALbus<x>:ASETup:EXECute
<x> = 1 to 4

Example :SERIALBUS1:ASETUP:EXECUTE

:SERIALbus<x>:ASETup:SWAVEform

Function Sets or queries the waveform on which to perform serial bus signal auto setup.

Syntax :SERIALbus<x>:ASETup:
SWAVEform {KEEP|UPDATE}
:SERIALbus<x>:ASETup:SWAVEform?
<x> = 1 to 4

Example :SERIALBUS1:ASETUP:SWAVEFORM KEEP
:SERIALBUS1:ASETUP:SWAVEFORM?
-> :SERIALBUS1:ASETUP:SWAVEFORM KEEP

Description • This command applies to SERIALbus1 to SERIALbus4.
• If KEEP is specified, auto setup is performed on the waveforms displayed on the screen.
• If UPDATE is specified, auto setup will be performed on new waveforms that will be acquired.

:SERIALbus<x>:ASETup:TSETing

Function Sets or queries whether trigger settings are included in the serial bus signal auto setup.

Syntax :SERIALbus<x>:ASETup:
TSETing {<boolean>}
:SERIALbus<x>:ASETup:TSETing?
<x> = 1 to 4

Example :SERIALBUS1:ASETUP:TSETING ON
:SERIALBUS1:ASETUP:TSETING?
-> :SERIALBUS1:ASETUP:TSETING 1

Description This command applies to SERIALbus1 to SERIALbus4.

:SERIALbus<x>:CAN?

Function Queries all CAN bus signal settings.

Syntax :SERIALbus<x>:CAN?
<x> = 1 to 4

:SERIALbus<x>:CAN:ANALyze?

Function Queries all CAN bus signal analysis settings.

Syntax :SERIALbus<x>:CAN:ANALyze?
<x> = 1 to 4

:SERIALbus<x>:CAN[:ANALyze]:SETup?

Function Queries all CAN bus signal analysis bus settings.

Syntax :SERIALbus<x>:CAN[:ANALyze]:SETup?
<x> = 1 to 4

:SERIALbus<x>:CAN[:ANALyze]:SETup:BRATe

Function Sets or queries the CAN bus signal analysis bit rate (data transfer rate).

Syntax :SERIALbus<x>:CAN[:ANALyze]:SETup:
BRATe {<Nrf>|USER,<Nrf>}
:SERIALbus<x>:CAN[:ANALyze]:SETup:
BRATe?
<x> = 1 to 4
<Nrf> = 33300, 83300, 125000, 250000,
500000, 1000000
USER <Nrf> = See the Features Guide for this
information.

Example :SERIALBUS1:CAN:ANALYZE:SETUP:
BRATE 83300
:SERIALBUS1:CAN:ANALYZE:SETUP:BRATE?
-> :SERIALBUS1:CAN:ANALYZE:SETUP:
BRATE 83300

:SERIALbus<x>:CAN[:ANALyze]:SETup:RECe ssive

Function Sets or queries the CAN bus signal analysis recessive level.

Syntax :SERIALbus<x>:CAN[:ANALyze]:SETup:
RECe ssive {HIGH|LOW}
:SERIALbus<x>:CAN[:ANALyze]:SETup:
RECe ssive?
<x> = 1 to 4

Example :SERIALBUS1:CAN:ANALYZE:SETUP:
RECESSIVE HIGH
:SERIALBUS1:CAN:ANALYZE:SETUP:
RECESSIVE?
-> :SERIALBUS1:CAN:ANALYZE:SETUP:
RECESSIVE HIGH

:SERIALbus<x>:CAN[:ANALyze]:SETup:SOUR ce

Function Sets or queries the CAN bus signal analysis source.

Syntax :SERIALbus<x>:CAN[:ANALyze]:SETup:
SOURce {<Nrf>|MATH<y>}
:SERIALbus<x>:CAN[:ANALyze]:SETup:
SOURce?
<x> = 1 to 4
<Nrf> = 1 to 4
MATH<y>'s <y> = 1 to 4

Example :SERIALBUS1:CAN:ANALYZE:SETUP:
SOURCE 1
:SERIALBUS1:CAN:ANALYZE:SETUP:
SOURCE?
-> :SERIALBUS1:CAN:ANALYZE:SETUP:
SOURCE 1

:SERIALbus<x>:CAN[:ANALyze]:SETup:SPOi nt

Function Sets or queries the CAN bus signal analysis sample point.

Syntax :SERIALbus<x>:CAN[:ANALyze]:SETup:
SPOi nt {<Nrf>}
:SERIALbus<x>:CAN[:ANALyze]:SETup:
SPOi nt?
<x> = 1 to 4
<Nrf> = 18.8 to 90.6

Example :SERIALBUS1:CAN:ANALYZE:SETUP:
SPOINT 18.8
:SERIALBUS1:CAN:ANALYZE:SETUP:
SPOINT?
-> :SERIALBUS1:CAN:ANALYZE:SETUP:
SPOINT 18.8

:SERIALbus<x>:CAN:DETAil?

Function Queries all CAN bus signal analysis result list settings.

Syntax :SERIALbus<x>:CAN:DETAil?
<x> = 1 to 4

:SERIALbus<x>:CAN:DETAil:DISPlay

Function Sets or queries the display mode for the CAN bus signal analysis result list.

Syntax :SERIALbus<x>:CAN:DETAil:
DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:CAN:DETAil:DISPlay?
<x> = 1 to 4

Example :SERIALBUS1:CAN:DETAIL:DISPLAY FULL
:SERIALBUS1:CAN:DETAIL:DISPLAY?
-> :SERIALBUS1:CAN:DETAIL:
DISPLAY FULL

5.24 SERIALbus group

:SERIALbus<x>:CAN:DETAil:LIST:ITEM?

Function Queries all items that will be displayed in the CAN bus signal analysis result list.

Syntax :SERIALbus<x>:CAN:DETAil:LIST:ITEM?
<x> = 1 to 4

Example :SERIALBUS1:CAN:DETAil:LIST:ITEM?
-> :SERIALBUS1:CAN:DETAil:LIST:
ITEM "No.,Time(ms),Frame,ID,DLC,
Data,CRC,Ack,Information"

:SERIALbus<x>:CAN:DETAil:LIST:VALue?

Function Queries all the data for the specified analysis number in the CAN bus signal analysis result list.

Syntax :SERIALbus<x>:CAN:DETAil:LIST:
VALue? {<NRf>|MAXimum|MINimum}
<x> = 1 to 4

<NRf> = See the Features Guide for this information.

Example :SERIALBUS1:CAN:DETAil:LIST:VALue? 0
-> :SERIALBUS1:CAN:DETAil:LIST:
VALue "0,0.0750,Data,012,1,FE,2263,
Y,, "

:SERIALbus<x>:CAN:SEARCh?

Function Queries all CAN bus signal search settings.

Syntax :SERIALbus<x>:CAN:SEARCh?
<x> = 1 to 4

:SERIALbus<x>:CAN:SEARCh:ABORT

Function Aborts the CAN bus signal search.

Syntax :SERIALbus<x>:CAN:SEARCh:ABORT
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCh:ABORT

:SERIALbus<x>:CAN:SEARCh:EXECute

Function Executes a CAN bus signal search.

Syntax :SERIALbus<x>:CAN:SEARCh:EXECute
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCh:EXECUTE

:SERIALbus<x>:CAN:SEARCh:FJUMp:ACK

Function Jumps to the ACK field in the CAN bus signal search result.

Syntax :SERIALbus<x>:CAN:SEARCh:FJUMp:ACK
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCh:FJUMp:ACK

:SERIALbus<x>:CAN:SEARCh:FJUMp:CONTRol

Function Jumps to the control field in the CAN bus signal search result.

Syntax :SERIALbus<x>:CAN:SEARCh:FJUMp:
CONTRol
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCh:FJUMp:CONTROL

:SERIALbus<x>:CAN:SEARCh:FJUMp:CRC

Function Jumps to the CRC field in the CAN bus signal search result.

Syntax :SERIALbus<x>:CAN:SEARCh:FJUMp:CRC
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCh:FJUMp:CRC

:SERIALbus<x>:CAN:SEARCh:FJUMp:DATA

Function Jumps to the data field in the CAN bus signal search result.

Syntax :SERIALbus<x>:CAN:SEARCh:FJUMp:DATA
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCh:FJUMp:DATA

:SERIALbus<x>:CAN:SEARCh:FJUMp:IDENTifier

Function Jumps to the identifier in the CAN bus signal search result.

Syntax :SERIALbus<x>:CAN:SEARCh:FJUMp:
IDENTifier
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCh:FJUMp:
IDENTIFIER

:SERIALbus<x>:CAN:SEARCh:FJUMp:SOF

Function Jumps to the SOF in the CAN bus signal search result.

Syntax :SERIALbus<x>:CAN:SEARCh:FJUMp:SOF
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCh:FJUMp:SOF

:SERIALbus<x>:CAN:SEARCh:SELEct

Function Sets which detected point to display in the CAN bus signal search zoom window and queries the zoom position of the detected point.

Syntax :SERIALbus<x>:CAN:SEARCh:SELEct {<NRf>|MAXimum}
:SERIALbus<x>:CAN:SEARCh:SELEct?<x> = 1 to 4<NRf> = 0 to 49999

Example :SERIALBUS1:CAN:SEARCH:SELECT 1
:SERIALBUS1:CAN:SEARCH:SELECT?
-> :SERIALBUS1:CAN:SEARCH:SELECT 1.50000000

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:CAN:SEARCh:SELEct? MAXimum

Function Queries the number of detected points in the CAN bus signal search.

Syntax :SERIALbus<x>:CAN:SEARCh:SELEct? {MAXimum}<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SELECT? MAXIMUM
-> :SERIALBUS1:CAN:SEARCH:SELECT 100

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:CAN:SEARCh:SETup?

Function Queries all CAN bus signal search condition settings.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup?<x> = 1 to 4

:SERIALbus<x>:CAN:SEARCh:SETup:EFrame?

Function Queries all CAN bus signal search error settings.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup:EFFrame?<x> = 1 to 4

:SERIALbus<x>:CAN:SEARCh:SETup:EFFrame[:MODE]

Function Sets or queries the CAN bus signal search error frame setting.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup:EFFrame[:MODE] {<Boolean>}<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:EFFrame:MODE ON
:SERIALBUS1:CAN:SEARCH:SETUP:EFFrame:MODE?
-> :SERIALBUS1:CAN:SEARCH:SETUP:EFFrame:MODE 1

:SERIALbus<x>:CAN:SEARCh:SETup:EFFrame:CRC

Function Sets or queries the CAN bus signal search CRC error setting.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup:EFFrame:CRC {<Boolean>}<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:EFFrame:CRC ON
:SERIALBUS1:CAN:SEARCH:SETUP:EFFrame:CRC?
-> :SERIALBUS1:CAN:SEARCH:SETUP:EFFrame:CRC 1

:SERIALbus<x>:CAN:SEARCh:SETup:EFFrame:STUFF

Function Sets or queries the CAN bus signal search stuff error setting.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup:EFFrame:STUFF {<Boolean>}<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:EFFrame:STUFF ON
:SERIALBUS1:CAN:SEARCH:SETUP:EFFrame:STUFF?
-> :SERIALBUS1:CAN:SEARCH:SETUP:EFFrame:STUFF 1

:SERIALbus<x>:CAN:SEARCh:SETup:IDData?

Function Queries all ID and data condition settings for CAN bus signal searching.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup:IDData?<x> = 1 to 4

5.24 SERIALbus group

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:ACK?

Function Queries all ID and data condition ACK settings for CAN bus signal searching.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:ACK?
<x> = 1 to 4

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:ACK:MODE

Function Sets or queries the ACK mode, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:ACK:MODE {<Boolean>}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:ACK:MODE?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
ACK:MODE ON
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
ACK:MODE?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:ACK:MODE 1

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:ACK:TYPE

Function Sets or queries the ACK condition, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:ACK:TYPE {ACK|ACKBoth|NONAck}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:ACK:TYPE?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
ACK:TYPE ACK
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
ACK:TYPE?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:ACK:TYPE ACK

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA?

Function Queries all data settings for the ID and data conditions for CAN bus signal searching.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA?
<x> = 1 to 4

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:CONDition

Function Sets or queries the data comparison condition, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:CONDition {EQUAL|FALSE|GREATER|INSide|LESS|NOTEQUAL|OUTSide|TRUE}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:CONDITION EQUAL
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:CONDITION?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:DATA:CONDITION EQUAL

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:DECimal<y>

Function Sets or queries a reference value, which is one of the ID and data conditions, for CAN bus signal searching in decimal notation.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:DECimal<y> {<Nrf>,<Nrf>}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:DECimal<y>?
<x> = 1 to 4
<y> = 1 to 2
<Nrf> = See the Features Guide for this information.

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:DECIMAL1 1
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:DECIMAL1?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1 1

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:DLC

Function Sets or queries the number of significant bytes (DLC), which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:DLC {<Nrf>}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:DLC?
<x> = 1 to 4
<Nrf> = 0 to 8

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:DLC 0
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:DLC?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:DATA:DLC 0

**:Serialbus<x>:CAN:SEARCh:SETup[:IDData]
]:DATA:DPATtern?**

Function Queries all judgment data settings for the ID and data conditions for CAN bus signal searching.

Syntax :Serialbus<x>:CAN:SEARCh:SETup[:
IDData]:DATA:DPATtern?
<x> = 1 to 4

**:Serialbus<x>:CAN:SEARCh:SETup[:IDData]
]:DATA:DPATtern:HEXa**

Function Sets a reference value, which is one of the ID and data conditions, for CAN bus signal searching in hexadecimal notation.

Syntax :Serialbus<x>:CAN:SEARCh:SETup[:
IDData]:DATA:DPATtern:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:DPATTERN:HEXA "12"

Description The number of bytes that can be set varies depending on the :Serialbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:DLC setting.

**:Serialbus<x>:CAN:SEARCh:SETup[:IDData]
]:DATA:DPATtern:PATtern**

Function Sets or queries the data, which is one of the ID and data conditions, for CAN bus signal searching in binary notation.

Syntax :Serialbus<x>:CAN:SEARCh:SETup[:
IDData]:DATA:DPATtern:
PATtern {<String>}
:Serialbus<x>:CAN:SEARCh:SETup[:
IDData]:DATA:DPATtern:PATtern?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:DPATTERN:PATTERN "00110101"
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:DPATTERN:PATTERN ?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:
PATTERN "00110101"

Description The number of bytes that can be set varies depending on the :Serialbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:DLC setting.

**:Serialbus<x>:CAN:SEARCh:SETup[:IDData]
]:DATA:ENDian**

Function Sets or queries the byte order of the reference values, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :Serialbus<x>:CAN:SEARCh:SETup[:
IDData]:DATA:ENDian {BIG|LITtle}
:Serialbus<x>:CAN:SEARCh:SETup[:
IDData]:DATA:ENDian?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:ENDIAN BIG
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:ENDIAN?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG

**:Serialbus<x>:CAN:SEARCh:SETup[:IDData]
]:DATA:MODE**

Function Sets or queries the data condition (enabled/disabled), which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :Serialbus<x>:CAN:SEARCh:SETup[:
IDData]:DATA:MODE {<Boolean>}
:Serialbus<x>:CAN:SEARCh:SETup[:
IDData]:DATA:MODE?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:MODE ON
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:MODE?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:DATA:MODE 1

**:Serialbus<x>:CAN:SEARCh:SETup[:IDData]
]:DATA:MSBLsb**

Function Sets or queries the MSB and LSB bits for the data, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :Serialbus<x>:CAN:SEARCh:SETup[:
IDData]:DATA:MSBLsb {<NRf>,<NRf>}
:Serialbus<x>:CAN:SEARCh:SETup[:
IDData]:DATA:MSBLsb?
<x> = 1 to 4
<NRf> = 0 to 63

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:MSBLSB 1,0
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:MSBLSB?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 1,0

5.24 SERIALbus group

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:SIGN

Function Sets or queries the sign, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:SIGN {SIGN|UNSign}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:DATA:SIGN?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:SIGN SIGN
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
DATA:SIGN?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier?

Function Queries the ID, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier?
<x> = 1 to 4

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:HEXa

Function Sets an ID, which is one of the ID and data conditions, for CAN bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
IDENTIFIER:HEXa "1AB"

Description The number of bytes that can be set varies depending on the :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:MFormat setting.

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:MFormat

Function Sets or queries the ID frame format (standard or extended), which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:
MFormat {EXTended|STANdard}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:MFormat?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
IDENTIFIER:MFORMAT EXTENDED
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
IDENTIFIER:MFORMAT?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:IDENTIFIER:MFORMAT EXTENDED

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:MODE

Function Sets or queries the ID condition (enabled/disabled), which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:MODE {<Boolean>}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:MODE?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
IDENTIFIER:MODE ON
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
IDENTIFIER:MODE?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:IDENTIFIER:MODE 1

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:PATtern

Function Sets or queries the ID condition, which is one of the ID and data conditions, for CAN bus signal searching in binary notation.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:PATtern{<String>}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:PATtern?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
IDENTIFIER:PATTERN "11100001111"
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
IDENTIFIER:PATTERN?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:IDENTIFIER:
PATTERN "11100001111"

Description The number of bytes that can be set varies depending on the :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:MFormat setting.

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:PFormat

Function Sets or queries the ID input format, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:PFormat {MESSAge|PATtern}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:IDENTifier:PFormat?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
IDENTIFIER:PFORMAT PATTERN
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
IDENTIFIER:PFORMAT?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:IDENTIFIER:PFORMAT PATTERN

:SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal?

Function Queries the message signal, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal?
<x> = 1 to 4

:SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:MESSAge:ITEM

Function Sets the message item, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:MESSAge:ITEM{<String>}
<x> = 1 to 4
<String> = Up to 32 characters

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:MESSAGE:ITEM "TEST"

:SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:SELEct

Function Sets or queries the message signal condition, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:SELEct {MESSAGE|SIGNAL}
:SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:SELEct?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:SELECT MESSAGE
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:SELECT?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:SELECT MESSAGE

:SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:SIGnal?

Function Queries the signal, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:SIGnal?
<x> = 1 to 4

:SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:SIGnal:CONDition

Function Sets or queries the signal data condition, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:SIGnal:CONDition {EQUAL|GREATER|INSIDE|LESS|NOTEQUAL|OUTSIDE}
:SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:SIGnal:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:SIGNAL:CONDITION EQUAL
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:SIGNAL:CONDITION?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:SIGNAL:CONDITION EQUAL

:SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:SIGnal:DECimal<y>

Function Sets or queries a reference signal value, which is one of the ID and data conditions, for CAN bus signal searching in decimal notation.

Syntax :SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:SIGnal:DECimal<y> {<NRF>}
:SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:SIGnal:DECimal<y>?
<x> = 1 to 4
<y> = 1 to 2
<NRF> = See the Features Guide for this information.

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:SIGNAL:DECIMAL1 1
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:SIGNAL:DECIMAL1?
-> :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:SIGNAL:DECIMAL1 1.000E+00

:SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:SIGnal:ITEM

Function Sets the signal item, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERialbus<x>:CAN:SEARCh:SETup[:IDData]:MSIGnal:SIGnal:ITEM {<String>,<String>}
<x> = 1 to 4
<String> = Up to 32 characters

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:MSIGNAL:SIGNAL:ITEM "Sig _ Test",
"Mess _ Test"

Description Set the signal item first and then the message item.

5.24 SERIALbus group

:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:RTR

Function Sets or queries the RTR value, which is one of the ID and data conditions, for CAN bus signal searching.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:RTR {<Boolean>}
:SERIALbus<x>:CAN:SEARCh:SETup[:IDData]:RTR?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
RTR ON
:SERIALBUS1:CAN:SEARCH:SETUP:IDDATA:
RTR?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
IDDATA:RTR 1

:SERIALbus<x>:CAN:SEARCh:SETup:MODE

Function Sets or queries the CAN bus signal search mode setting.

Syntax :SERIALbus<x>:CAN:SEARCh:SETup:
MODE {EFrame|IDData|SOF}
:SERIALbus<x>:CAN:SEARCh:SETup:MODE?
<x> = 1 to 4

Example :SERIALBUS1:CAN:SEARCH:SETUP:
MODE EFRAME
:SERIALBUS1:CAN:SEARCH:SETUP:MODE?
-> :SERIALBUS1:CAN:SEARCH:SETUP:
MODE EFRAME

:SERIALbus<x>:CANFD?

Function Queries all CAN FD bus signal settings.

Syntax :SERIALbus<x>:CANFD?
<x> = 1 to 4

:SERIALbus<x>:CANFD:ANALyze?

Function Queries all CAN FD bus signal analysis settings.

Syntax :SERIALbus<x>:CANFD:ANALyze?
<x> = 1 to 4

:SERIALbus<x>:CANFD[:ANALyze]:SETup?

Function Queries all CAN FD bus signal analysis bus settings.

Syntax :SERIALbus<x>:CANFD[:ANALyze]:SETup?
<x> = 1 to 4

:SERIALbus<x>:CANFD[:ANALyze]:SETup:BRATe

Function Sets or queries the CAN FD bus signal analysis bit rate (data transfer rate).

Syntax :SERIALbus<x>:CANFD[:ANALyze]:SETup:
BRATe {<Nrf>|USER,<Nrf>}
:SERIALbus<x>:CANFD[:ANALyze]:SETup:
BRATe?
<x> = 1 to 4
<Nrf> = 250000, 500000, 1000000
USER <Nrf> = 20000 to 1000000 (resolution:
100)

Example :SERIALBUS1:CANFD:ANALYZE:SETUP:
BRATE 1000000
:SERIALBUS1:CANFD:ANALYZE:SETUP:
BRATE?
-> :SERIALBUS1:CANFD:ANALYZE:SETUP:
BRATE 1000000

:SERIALbus<x>:CANFD[:ANALyze]:SETup:DBRate

Function Sets or queries the CAN FD bus signal analysis data bit rate (data transfer rate of data phase).

Syntax :SERIALbus<x>:CANFD[:ANALyze]:SETup:
DBRate {<Nrf>|USER,<Nrf>}
:SERIALbus<x>:CANFD[:ANALyze]:SETup:
DBRate?
<x> = 1 to 4
<Nrf> = 500000, 1000000, 2000000, 4000000,
5000000, 8000000
USER <Nrf> = 250000 to 10000000 (resolution:
100)

Example :SERIALBUS1:CANFD:ANALYZE:SETUP:
DBRATE 5000000
:SERIALBUS1:CANFD:ANALYZE:SETUP:
DBRATE?
-> :SERIALBUS1:CANFD:ANALYZE:SETUP:
DBRATE 5000000

:SERIALbus<x>:CANFD[:ANALyze]:SETup:DSPoint

Function Sets or queries the CAN FD bus signal analysis data phase sample point.

Syntax :SERIALbus<x>:CANFD[:ANALyze]:SETup:DSPoint {<Nrf>}
:SERIALbus<x>:CANFD[:ANALyze]:SETup:DSPoint?
<x> = 1 to 4
<Nrf> = 18.8 to 90.6 (resolution: 0.1)

Example :SERIALBUS1:CANFD:ANALYZE:SETUP:DSPOINT 75
:SERIALBUS1:CANFD:ANALYZE:SETUP:DSPOINT?
-> :SERIALBUS1:CANFD:ANALYZE:SETUP:DSPOINT 75.00000000

:SERIALbus<x>:CANFD[:ANALyze]:SETup:FDStandard

Function Sets or queries whether the CAN FD bus signal to be analyzed is an ISO standard signal.

Syntax :SERIALbus<x>:CANFD[:ANALyze]:SETup:FDStandard {ISO|NISO}
:SERIALbus<x>:CANFD[:ANALyze]:SETup:FDStandard?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:ANALYZE:SETUP:FDSTANDARD ISO
:SERIALBUS1:CANFD:ANALYZE:SETUP:FDSTANDARD?
-> :SERIALBUS1:CANFD:ANALYZE:SETUP:FDSTANDARD ISO

:SERIALbus<x>:CANFD[:ANALyze]:SETup:REcessive

Function Sets or queries the CAN FD bus signal analysis recessive level.

Syntax :SERIALbus<x>:CANFD[:ANALyze]:SETup:REcessive {HIGH|LOW}
:SERIALbus<x>:CANFD[:ANALyze]:SETup:REcessive?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:ANALYZE:SETUP:RECESSIVE HIGH
:SERIALBUS1:CANFD:ANALYZE:SETUP:RECESSIVE?
-> :SERIALBUS1:CANFD:ANALYZE:SETUP:RECESSIVE HIGH

:SERIALbus<x>:CANFD[:ANALyze]:SETup:SOURce

Function Sets or queries the CAN FD bus signal analysis source.

Syntax :SERIALbus<x>:CANFD[:ANALyze]:SETup:SOURce {<Nrf>|MATH<y>}
:SERIALbus<x>:CANFD[:ANALyze]:SETup:SOURce?
<x> = 1 to 4
<Nrf> = 1 to 4
MATH<y>'s <y> = 1 to 4

Example :SERIALBUS1:CANFD:ANALYZE:SETUP:SOURCE 1
:SERIALBUS1:CANFD:ANALYZE:SETUP:SOURCE?
-> :SERIALBUS1:CANFD:ANALYZE:SETUP:SOURCE 1

:SERIALbus<x>:CANFD[:ANALyze]:SETup:SPOint

Function Sets or queries the CAN FD bus signal analysis sample point.

Syntax :SERIALbus<x>:CANFD[:ANALyze]:SETup:SPOint {<Nrf>}
:SERIALbus<x>:CANFD[:ANALyze]:SETup:SPOint?
<x> = 1 to 4
<Nrf> = 18.8 to 90.6 (resolution: 0.1)

Example :SERIALBUS1:CANFD:ANALYZE:SETUP:SPOINT 75
:SERIALBUS1:CANFD:ANALYZE:SETUP:SPOINT?
-> :SERIALBUS1:CANFD:ANALYZE:SETUP:SPOINT 75

:SERIALbus<x>:CANFD:DETail?

Function Queries all CAN FD bus signal analysis result list settings.

Syntax :SERIALbus<x>:CANFD:DETail?
<x> = 1 to 4

:SERIALbus<x>:CANFD:DETail:DISPlay

Function Sets or queries the display mode for the CAN FD bus signal analysis result list.

Syntax :SERIALbus<x>:CANFD:DETail:DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:CANFD:DETail:DISPlay
<x> = 1 to 4

Example :SERIALBUS1:CANFD:DETAIL:DISPLAY FULL
:SERIALBUS1:CANFD:DETAIL:DISPLAY?
-> :SERIALBUS1:CANFD:DETAIL:DISPLAY FULL

5.24 SERIALbus group

:SERIALbus<x>:CANFD:DETAil:LIST:ITEM?
Function Queries all items that will be displayed in the CAN FD bus signal analysis result list.
Syntax :SERIALbus<x>:CANFD:DETAil:LIST:ITEM?
<x> = 1 to 4
Example :SERIALBUS1:CANFD:DETAIL:LIST:ITEM?
-> :SERIALBUS1:CANFD:DETAIL:LIST:ITEM "No.,Time(ms),Frame,ID,DLC,Data,CRC,Ack,Information"

:SERIALbus<x>:CANFD:DETAil:LIST:VALue?
Function Queries all the data for the specified analysis number in the CAN FD bus signal analysis result list.
Syntax :SERIALbus<x>:CANFD:DETAil:LIST:VALue? {<NRF>|MAXimum|MINimum}
<x> = 1 to 4
<NRF> = -49999 to 49999
Example :SERIALBUS1:CANFD:DETAIL:LIST:VALue? 0
-> :SERIALBUS1:CANFD:DETAIL:LIST:VALUE "0,0.0750,Data,012,1,FE,2263,Y,,"

:SERIALbus<x>:CANFD:SEARCh?
Function Queries all CAN FD bus signal search settings.
Syntax :SERIALbus<x>:CANFD:SEARCh?
<x> = 1 to 4

:SERIALbus<x>:CANFD:SEARCh:ABORT
Function Aborts the CAN FD bus signal search.
Syntax :SERIALbus<x>:CANFD:SEARCh:ABORT
<x> = 1 to 4
Example :SERIALBUS1:CANFD:SEARCH:ABORT

:SERIALbus<x>:CANFD:SEARCh:EXECute
Function Executes a CAN FD bus signal search.
Syntax :SERIALbus<x>:CANFD:SEARCh:EXECute
<x> = 1 to 4
Example :SERIALBUS1:CANFD:SEARCH:EXECUTE

:SERIALbus<x>:CANFD:SEARCh:FJUMp:ACK
Function Jumps to the ACKField in the CAN FD bus signal search result.
Syntax :SERIALbus<x>:CANFD:SEARCh:FJUMp:ACK
<x> = 1 to 4
Example :SERIALBUS1:CANFD:SEARCH:FJUMP:ACK

:SERIALbus<x>:CANFD:SEARCh:FJUMp:CONTrOl
Function Jumps to the control field in the CAN FD bus signal search result.
Syntax :SERIALbus<x>:CANFD:SEARCh:FJUMp:CONTrOl
<x> = 1 to 4
Example :SERIALBUS1:CANFD:SEARCH:FJUMP:CONTROL

:SERIALbus<x>:CANFD:SEARCh:FJUMp:CRC
Function Jumps to the CRC field in the CAN FD bus signal search result.
Syntax :SERIALbus<x>:CANFD:SEARCh:FJUMp:CRC
<x> = 1 to 4
Example :SERIALBUS1:CANFD:SEARCH:FJUMP:CRC

:SERIALbus<x>:CANFD:SEARCh:FJUMp:DATA
Function Jumps to the data field in the CAN FD bus signal search result.
Syntax :SERIALbus<x>:CANFD:SEARCh:FJUMp:DATA
<x> = 1 to 4
Example :SERIALBUS1:CANFD:SEARCH:FJUMP:DATA

:SERIALbus<x>:CANFD:SEARCh:FJUMp:IDENtifier
Function Jumps to the identifier in the CAN FD bus signal search result.
Syntax :SERIALbus<x>:CANFD:SEARCh:FJUMp:IDENtifier
<x> = 1 to 4
Example :SERIALBUS1:CANFD:SEARCH:FJUMP:IDENTIFIER

:SERIALbus<x>:CANFD:SEARCh:FJUMp:SOF
Function Jumps to the SOF in the CAN FD bus signal search result.
Syntax :SERIALbus<x>:CANFD:SEARCh:FJUMp:SOF
<x> = 1 to 4
Example :SERIALBUS1:CANFD:SEARCH:FJUMP:SOF

:SERIALbus<x>:CANFD:SEARCh:SELEct

Function Sets which detected point to display in the CAN FD bus signal search zoom window and queries the zoom position of the detected point.

Syntax :SERIALbus<x>:CANFD:SEARCh:
SELEct {<NRf>|MAXimum}
:SERIALbus<x>:CANFD:SEARCh:SELEct?
<x> = 1 to 4
<NRf> = 0 to 49999

Example :SERIALBUS1:CANFD:SEARCH:SELECT 1
:SERIALBUS1:CANFD:SEARCH:SELECT?
-> :SERIALBUS1:CANFD:SEARCH:
SELECT 1.50000000

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:CANFD:SEARCh:SELEct? MAXimum

Function Queries the number of detected points in the CAN FD bus signal search.

Syntax :SERIALbus<x>:CANFD:SEARCh:
SELEct? {MAXimum}
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:
SELECT? MAXIMUM
-> :SERIALBUS1:CANFD:SEARCH:
SELECT 100

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:CANFD:SEARCh:SETup?

Function Queries all CAN FD bus signal search condition settings.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup?
<x> = 1 to 4

:SERIALbus<x>:CANFD:SEARCh:SETup:EFramE?

Function Queries all CAN FD bus signal search error settings.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup:
EFramE?
<x> = 1 to 4

:SERIALbus<x>:CANFD:SEARCh:SETup:EFramE:CRC

Function Sets or queries the CAN FD bus signal search CRC error setting.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup:
EFramE:CRC {<Boolean>}
:SERIALbus<x>:CANFD:SEARCh:SETup:
EFramE:CRC?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRC ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRC?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRC 1

:SERIALbus<x>:CANFD:SEARCh:SETup:EFramE:CRCEFactor?

Function Queries all CRC error factor settings for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup:
EFramE:CRCEFactor?
<x> = 1 to 4

:SERIALbus<x>:CANFD:SEARCh:SETup:EFramE:CRCEFactor:CRCSequence

Function Sets or queries the CRC sequence (a CRC error factor) for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup:
EFramE:CRCEFactor:
CRCSequence {<Boolean>}
:SERIALbus<x>:CANFD:SEARCh:SETup:
EFramE:CRCEFactor:CRCSequence?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRCFactor:CRCSEQUENCE ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRCFactor:CRCSEQUENCE?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRCFactor:CRCSEQUENCE 1

5.24 SERIALbus group

:SERIALbus<x>:CANFD:SEARCh:SETup:EFra me:CRCEFactor:SCount

Function Sets or queries the Stuff Count (a CRC error factor) for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup:
EFrame:CRCEFactor:SCount {<Boolean>}
:SERIALbus<x>:CANFD:SEARCh:SETup:
EFrame:CRCEFactor:SCount?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRCEFACTOR:SCOUNT ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRCEFACTOR:SCOUNT?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:CRCEFACTOR:SCOUNT 1

:SERIALbus<x>:CANFD:SEARCh:SETup:EFra me:FSTuff

Function Sets or queries the CAN FD bus signal search fixed stuff error setting.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup:
EFrame:FSTuff {<Boolean>}
:SERIALbus<x>:CANFD:SEARCh:SETup:
EFrame:FSTuff?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:FSTUFF ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:FSTUFF?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:FSTUFF 1

:SERIALbus<x>:CANFD:SEARCh:SETup:EFra me[:MODE]

Function Sets or queries the CAN FD bus signal search error frame setting.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup:
EFrame[:MODE] {<Boolean>}
:SERIALbus<x>:CANFD:SEARCh:SETup:
EFrame[:MODE]?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:MODE ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:MODE?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:MODE 1

:SERIALbus<x>:CANFD:SEARCh:SETup:EFra me:STUFF

Function Sets or queries the CAN FD bus signal search stuff error setting.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup:
EFrame:STUFF {<Boolean>}
:SERIALbus<x>:CANFD:SEARCh:SETup:
EFrame:STUFF?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:STUFF ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:STUFF?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
EFRAME:STUFF 1

:SERIALbus<x>:CANFD:SEARCh:SETup:FDF:C ONDITION

Function Sets or queries the CAN FD bus signal search FDF condition setting.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup:
FDF:CONDition {<Boolean>}
:SERIALbus<x>:CANFD:SEARCh:SETup:
FDF:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:FDF:
CONDITION ON
:SERIALBUS1:CANFD:SEARCH:SETUP:FDF:
CONDITION?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
FDF:CONDITION 1

Description When set to ON, CAN FD frames are searched.
When set to OFF, CAN frames are searched.

:SERIALbus<x>:CANFD:SEARCh:SETup:IDDa ta?

Function Queries all ID and data condition settings for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup:
IDData?
<x> = 1 to 4

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDDa ta]:ACK?

Function Queries all ACK settings for the ID and data conditions for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:
IDData]:ACK?
<x> = 1 to 4

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:ACK:MODE

Function Sets or queries the ACK mode, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:ACK:MODE {<Boolean>}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:ACK:MODE?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:ACK:MODE ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:ACK:MODE?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:ACK:MODE 1

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:ACK:TYPE

Function Sets or queries the ACK condition, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:ACK:TYPE {ACK|ACKBoth|NONack}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:ACK:TYPE?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:ACK:TYPE ACK
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:ACK:TYPE?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:ACK:TYPE ACK

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA?

Function Queries all data settings for the ID and data conditions for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA?
<x> = 1 to 4

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:BCount

Function Sets or queries the pattern comparison position, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:BCOUNT {<Nrf>}
<x> = 1 to 4
<Nrf> = 0 to 63

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:BCOUNT 4
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:BCOUNT?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:BCOUNT 4

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:CONDition

Function Sets or queries the data comparison condition, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:CONDition {EQUAL|FALSE|GREater|INSide|LESS|NOTEqual|OUTSide|TRUE}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:CONDITION EQUAL
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:CONDITION?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:CONDITION EQUAL

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:DBYTe

Function Sets or queries the number of data bytes, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:DBYTe {<Nrf>}
<x> = 1 to 4
<Nrf> = 0 to 8

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DBYTE 2
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DBYTE?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DBYTE 2

5.24 SERIALbus group

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:DECimal<y>

Function Sets or queries a reference value, which is one of the ID and data conditions, for CAN FD bus signal searching in decimal notation.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:DECimal<y> {<NRF>}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:DECimal<y>?
<x> = 1 to 4
<y> = 1, 2
<NRF> = See the Features Guide for this information.

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1 1000
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1 1.0000000E+03

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:DPATtern?

Function Queries all judgment data settings for the ID and data conditions for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:DPATtern?
<x> = 1 to 4

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:DPATtern:HEXa

Function Sets a reference value, which is one of the ID and data conditions, for CAN FD bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:DPATtern:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:HEXA "12"

Description The number of bytes that can be set varies depending on the :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:DBYTe setting.

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:DPATtern:PATtern

Function Sets or queries the data value, which is one of the ID and data conditions, for CAN FD bus signal searching in binary notation.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:DPATtern:
PATtern {<String>}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:DPATtern:PATtern?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:
PATTERN "00110101"
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:PATTERN?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:
PATTERN "00110101"

Description The number of bytes that can be set varies depending on the :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:DBYTe setting.

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:ENDian

Function Sets or queries the byte order of the reference values, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:ENDian {BIG|LITtle}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:ENDian?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:ENDIAN?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:MODE

Function Sets or queries the data enable/disable condition, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:MODE {<Boolean>}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:MODE?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:MODE ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:MODE?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:MODE 1

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:MSBLSb

Function Sets or queries the MSB and LSB bits for the data, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:MSBLSb {<NRf>,<NRf>}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:MSBLSb?
<x> = 1 to 4
<NRf> = 0 to 63

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 15,0
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:MSBLSB?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 15,0

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:SIGN

Function Sets or queries whether signed or unsigned data format will be used, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:SIGN {SIGN|UNSign}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:DATA:SIGN?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:SIGN?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier?

Function Queries all identifier settings for the ID and data conditions for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier?
<x> = 1 to 4

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier:HEXa

Function Sets the ID value, which is one of the ID and data conditions, for CAN FD bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:HEXA "1AB"

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier:MFormat

Function Sets or queries the ID frame format (standard or extended), which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier:
MFormat {EXTended|STANdard}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier:MFormat?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:MFORMAT EXTENDED
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:MFORMAT?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:MFORMAT EXTENDED

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier:MODE

Function Sets or queries the ID enable/disable condition, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier:MODE {<Boolean>}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier:MODE?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:MODE ON
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:MODE?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:MODE 1

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier:PATtern

Function Sets or queries the ID pattern, which is one of the ID and data conditions, for CAN FD bus signal searching in binary notation.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier:PATtern {<String>}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier:PATtern?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:
PATTERN "11100001111"
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:PATTERN?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:
PATTERN "11100001111"

5.24 SERIALbus group

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier:PFormat

Function Sets or queries the ID input format, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier:PFormat {MESSagE|PATTern}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:IDENTifier:PFormat?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:PFORMAT PATTERN
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:PFORMAT?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:IDENTIFIER:PFORMAT PATTERN

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL?

Function Queries all message and signal settings for the ID and data conditions for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL?
<x> = 1 to 4

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:MESSagE:ITEM

Function Sets the message item, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:MESSagE:
ITEM {<String>}
<x> = 1 to 4
<String> = Up to 32 characters

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:MESSAGE:ITEM "TEST"

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:SElect

Function Sets or queries the message and signal condition, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:SElect {MESSagE|SIGNal}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:SElect?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SELECT MESSAGE
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SELECT?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SELECT MESSAGE

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:SIGNal?

Function Queries all signal settings for the ID and data conditions for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:SIGNal?
<x> = 1 to 4

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:SIGNal:CONDition

Function Sets or queries the signal data condition, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:SIGNal:
CONDition {EQUAL|GREATER|INSide|LESS|NOTEQUAL|OUTSide}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:SIGNal:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SIGNAL:
CONDITION EQUAL
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SIGNAL:CONDITION?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SIGNAL:
CONDITION EQUAL

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:SIGNal:DECimal<y>

Function Sets or queries a signal's reference value, which is one of the ID and data conditions, for CAN FD bus signal searching in decimal notation.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:SIGNal:
DECimal<y> {<NRf>}
:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:SIGNal:DECimal<y>?
<x> = 1 to 4
<y> = 1 to 2
<NRf> = See the Features Guide for this information.

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SIGNAL:DECIMAL1 1000
:SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SIGNAL:DECIMAL1?
-> :SERIALBUS1:CANFD:SEARCH:SETUP:
IDDATA:MSIGNAL:SIGNAL:
DECIMAL1 1.0000000E+03

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:SIGNAL:ITEM

Function Sets the signal item, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:MSIGNAL:SIGNAL:ITEM {<String>, <String>}
 <x> = 1 to 4
 <String> = Up to 32 characters

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
 IDDATA:MSIGNAL:SIGNAL:
 ITEM "Sig _ Test", "Mess _ Test"

Description Set the signal item first and then the message item.

:SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:RTR

Function Sets or queries the RTR value, which is one of the ID and data conditions, for CAN FD bus signal searching.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:RTR {<Boolean>}
 :SERIALbus<x>:CANFD:SEARCh:SETup[:IDData]:RTR?
 <x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
 IDDATA:RTR ON
 :SERIALBUS1:CANFD:SEARCH:SETUP:
 IDDATA:RTR?
 -> :SERIALBUS1:CANFD:SEARCH:SETUP:
 IDDATA:RTR 1

:SERIALbus<x>:CANFD:SEARCh:SETup:MODE

Function Sets or queries the CAN FD bus signal search mode setting.

Syntax :SERIALbus<x>:CANFD:SEARCh:SETup:MODE {EFrame|ESI|FDF|IDData|SOF}
 :SERIALbus<x>:CANFD:SEARCh:SETup:MODE?
 <x> = 1 to 4

Example :SERIALBUS1:CANFD:SEARCH:SETUP:
 MODE EFRAME
 :SERIALBUS1:CANFD:SEARCH:SETUP:MODE?
 -> :SERIALBUS1:CANFD:SEARCH:SETUP:
 MODE EFRAME

:SERIALbus<x>:CXPI?

Function Queries all CXPI bus signal analysis and search settings.

Syntax :SERIALbus<x>:CXPI?
 <x> = 1 to 4

:SERIALbus<x>:CXPI:ANALyze?

Function Queries all CXPI bus signal analysis settings.

Syntax :SERIALbus<x>:CXPI:ANALyze?
 <x> = 1 to 4

:SERIALbus<x>:CXPI[:ANALyze]:SETup?

Function Queries all CXPI bus signal analysis bus settings.

Syntax :SERIALbus<x>:CXPI[:ANALyze]:SETup?
 <x> = 1 to 4

:SERIALbus<x>:CXPI[:ANALyze]:SETup:BRATe

Function Sets or queries the CXPI bus signal analysis bit rate (data transfer rate).

Syntax :SERIALbus<x>:CXPI[:ANALyze]:SETup:BRATe {<Nrf>|USER,<Nrf>}
 :SERIALbus<x>:CXPI[:ANALyze]:SETup:BRATe?
 <x> = 1 to 4
 <Nrf> = 4800, 9600, 19200
 USER <Nrf> = 4000 to 50000

Example :SERIALBUS1:CXPI:ANALYZE:SETUP:
 BRATE 19200
 :SERIALBUS1:CXPI:ANALYZE:SETUP:
 BRATE?
 -> :SERIALBUS1:CXPI:ANALYZE:SETUP:
 BRATE 19200

:SERIALbus<x>:CXPI[:ANALyze]:SETup:CEDEtECTION

Function Sets or queries the enable/disable condition of counter error detection for CXPI bus signal analysis.

Syntax :SERIALbus<x>:CXPI[:ANALyze]:SETup:CEDEtECTION {<Boolean>}
 :SERIALbus<x>:CXPI[:ANALyze]:SETup:CEDEtECTION?
 <x> = 1 to 4

Example :SERIALBUS1:CXPI:ANALYZE:SETUP:
 CEDETECTION ON
 :SERIALBUS1:CXPI:ANALYZE:SETUP:
 CEDETECTION?
 -> :SERIALBUS1:CXPI:ANALYZE:SETUP:
 CEDETECTION 1

5.24 SERIALbus group

:SERIALbus<x>:CXPI[:ANALyze]:SETup:CTOLerance

Function Sets or queries the CXPI bus signal analysis clock tolerance.

Syntax :SERIALbus<x>:CXPI[:ANALyze]:SETup:CTOLerance {<Nrf>}
:SERIALbus<x>:CXPI[:ANALyze]:SETup:CTOLerance?
<x> = 1 to 4

Example <Nrf> = $\pm 0.5\%$ to $\pm 5.0\%$, resolution: 0.1%
:SERIALBUS1:CXPI:ANALYZE:SETUP:CTOLERANCE 5
:SERIALBUS1:CXPI:ANALYZE:SETUP:CTOLERANCE?
-> :SERIALBUS1:CXPI:ANALYZE:SETUP:CTOLERANCE 5.00E+00

:SERIALbus<x>:CXPI[:ANALyze]:SETup:SOURce

Function Sets or queries the CXPI bus signal analysis source.

Syntax :SERIALbus<x>:CXPI[:ANALyze]:SETup:SOURce {<Nrf>|MATH<y>}
:SERIALbus<x>:CXPI[:ANALyze]:SETup:SOURce?
<x> = 1 to 4
<y> = 1 to 4
<Nrf> = 1 to 4

Example :SERIALBUS1:CXPI:ANALYZE:SETUP:SOURCE 1
:SERIALBUS1:CXPI:ANALYZE:SETUP:SOURCE?
-> :SERIALBUS1:CXPI:ANALYZE:SETUP:SOURCE 1

:SERIALbus<x>:CXPI[:ANALyze]:SETup:TSAmpLe

Function Sets or queries the logic value (1 or 0) determination threshold for CXPI bus signal analysis.

Syntax :SERIALbus<x>:CXPI[:ANALyze]:SETup:TSAmpLe {<Nrf>}
<x> = 1 to 4
<Nrf> = 0.01 Tbit to 0.3 Tbit, resolution: 0.001 Tbit

Example :SERIALBUS1:CXPI:ANALYZE:SETUP:TSAMPLE 0.04
:SERIALBUS1:CXPI:ANALYZE:SETUP:TSAMPLE?
-> :SERIALBUS1:CXPI:ANALYZE:SETUP:TSAMPLE 0.04

Description For details on the logic value (1 or 0) determination threshold, see T Sample of CXPI analysis in the User's Manual.

:SERIALbus<x>:CXPI:DETail?

Function Queries all CXPI bus signal analysis result list settings.

Syntax :SERIALbus<x>:CXPI:DETail?
<x> = 1 to 4

:SERIALbus<x>:CXPI:DETail:DISPlay

Function Sets or queries the display mode for the CXPI bus signal analysis result list.

Syntax :SERIALbus<x>:CXPI:DETail:DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:CXPI:DETail:DISPlay?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:DETAIL:DISPLAY FULL
:SERIALBUS1:CXPI:DETAIL:DISPLAY?
-> :SERIALBUS1:CXPI:DETAIL:DISPLAY FULL

:SERIALbus<x>:CXPI:DETail:LIST:ALL?

Function Queries all the data for all analysis numbers in the CXPI bus signal analysis result list.

Syntax :SERIALbus<x>:CXPI:DETail:LIST:ALL?
<x> = 1 to 4

Example :SERIALBUS<x>:CXPI:DETAIL:LIST:ALL?
-> #8 (8-digit number of bytes)(data sequence)

Description All the data for all analysis numbers are returned in <block data> format. The data of each analysis number is separated by ASCII code 0AH.

:SERIALbus<x>:CXPI:DETail:LIST:ITEM?

Function Queries all items that will be displayed in the CXPI bus signal analysis result list.

Syntax :SERIALbus<x>:CXPI:DETail:LIST:ITEM?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:DETAIL:LIST:ITEM?
-> :SERIALBUS1:CXPI:DETAIL:LIST:ITEM "No.,Time(ms),ID,DLC,W/S,CT,Data,CRC,Information"

:SERIALbus<x>:CXPI:DETail:LIST:VALue?

Function Queries all the data for the specified analysis number in the CXPI bus signal analysis result list.

Syntax :SERIALbus<x>:CXPI:DETail:LIST:VALue? {<Nrf>|MAXimum|MINimum}
<x> = 1 to 4
<Nrf> = -9999 to 9999

Example :SERIALBUS1:CXPI:DETAIL:LIST:VALUE? 1
-> "1,4.7228,P3,8,01,0,00 00 00 00 01 03 05 06,FE,,,"

:SERIALbus<x>:CXPI:SEARCH?

Function Queries all CXPI bus signal search settings.

Syntax :SERIALbus<x>:CXPI:SEARCH?
<x> = 1 to 4

:SERIALbus<x>:CXPI:SEARCH:ABORT

Function Aborts the CXPI bus signal search.

Syntax :SERIALbus<x>:CXPI:SEARCH:ABORT
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:ABORT

:SERIALbus<x>:CXPI:SEARCH:EXECute

Function Executes a CXPI bus signal search.

Syntax :SERIALbus<x>:CXPI:SEARCH:EXECute
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:EXECUTE

:SERIALbus<x>:CXPI:SEARCH:SELEct

Function Sets which detected point to display in the CXPI bus signal search zoom window and queries the zoom position of the detected point.

Syntax :SERIALbus<x>:CXPI:SEARCH:
SELEct {<NRf>|MAXimum}
:SERIALbus<x>:CXPI:SEARCH:SELEct?
<x> = 1 to 4
<NRf> = 0 to 9999

Example :SERIALBUS1:CXPI:SEARCH:SELECT 1
:SERIALBUS1:CXPI:SEARCH:SELECT?
-> :SERIALBUS1:CXPI:SEARCH:
SELECT 1.50000000

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:CXPI:SEARCH:SETup?

Function Queries all CXPI bus signal search condition settings.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup?
<x> = 1 to 4

:SERIALbus<x>:CXPI:SEARCH:SETup:ERROR?

Function Queries all CXPI bus signal search error settings.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR?
<x> = 1 to 4

:SERIALbus<x>:CXPI:SEARCH:SETup:ERROR:CLOCK

Function Sets or queries the CXPI bus signal search clock error setting.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR:CLOCK {<Boolean>}
:SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR:CLOCK?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
CLOCK ON
:SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
CLOCK?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
ERROR:CLOCK 1

:SERIALbus<x>:CXPI:SEARCH:SETup:ERROR:COUNTER

Function Sets or queries the CXPI bus signal search counter error setting.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR:COUNTER {<Boolean>}
:SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR:COUNTER?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
COUNTER ON
:SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
COUNTER?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
ERROR:COUNTER 1

:SERIALbus<x>:CXPI:SEARCH:SETup:ERROR:CRC

Function Sets or queries the CXPI bus signal search CRC error setting.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR:CRC {<Boolean>}
:SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR:CRC?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
CRC ON
:SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
CRC?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
ERROR:CRC 1

5.24 SERIALbus group

:SERIALbus<x>:CXPI:SEARCH:SETup:ERROR:DLength

Function Sets or queries the CXPI bus signal search data length error setting.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR:DLength {<Boolean>}
:SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR:DLength?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
DLENGTH ON
:SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
DLENGTH?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
ERROR:DLENGTH 1

:SERIALbus<x>:CXPI:SEARCH:SETup:ERROR:FRAMing

Function Sets or queries the CXPI bus signal search framing error setting.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR:FRAMing {<Boolean>}
:SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR:FRAMing?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
FRAMING ON
:SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
FRAMING?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
ERROR:FRAMING 1

:SERIALbus<x>:CXPI:SEARCH:SETup:ERROR:IBS

Function Sets or queries the CXPI bus signal search IBS error setting.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR:IBS {<Boolean>}
:SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR:IBS?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
IBS ON
:SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
IBS?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
ERROR:IBS 1

:SERIALbus<x>:CXPI:SEARCH:SETup:ERROR:PARity

Function Sets or queries the CXPI bus signal search parity error setting.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR:PARity {<Boolean>}
:SERIALbus<x>:CXPI:SEARCH:SETup:
ERROR:PARity?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
PARITY ON
:SERIALBUS1:CXPI:SEARCH:SETUP:ERROR:
PARITY?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
ERROR:PARITY 1

:SERIALbus<x>:CXPI:SEARCH:SETup:IDData?

Function Queries all ID and data condition settings for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
IDData?
<x> = 1 to 4

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA?

Function Queries all data settings for the ID and data conditions for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:
IDData]:DATA?
<x> = 1 to 4

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:BCount

Function Sets or queries the pattern comparison position, which is one of the ID and data conditions, for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:
IDData]:DATA:BCount {<Nrf>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:
IDData]:DATA:BCount?

<x> = 1 to 4
<Nrf> = 0 to 254

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:BCOUNT 1
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:BCOUNT?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:BCOUNT 1

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:CONDition

Function Sets or queries the data comparison condition, which is one of the ID and data conditions, for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:CONDition {EQUAL|FALSE|GREATER|INSIDE|LESS|NOTEQUAL|OUTSIDE|TRUE}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:CONDITION EQUAL
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:CONDITION?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:CONDITION EQUAL

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DBYTE

Function Sets or queries the number of data bytes, which is one of the ID and data conditions, for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DBYTE {<Nrf>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DBYTE?
<x> = 1 to 4
<Nrf> = 0 to 8

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DBYTE 1
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DBYTE?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DBYTE 1

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DECimal<y>

Function Sets or queries a reference value, which is one of the ID and data conditions, for CXPI bus signal searching in decimal notation.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DECimal<y> {<Nrf>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DECimal<y>?
<x> = 1 to 4
<y> = 1 to 2
<Nrf> = See the Features Guide for this information.

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1 1
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1 1.0000000E+00

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DPATtern?

Function Queries all judgment data settings for the ID and data conditions for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DPATtern?
<x> = 1 to 4

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DPATtern:HEXa

Function Sets a reference value, which is one of the ID and data conditions, for CXPI bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DPATtern:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:HEXa "12"

Description The number of bytes that can be set varies depending on the :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DBYTE setting.

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DPATtern:PATtern

Function Sets or queries a reference value, which is one of the ID and data conditions, for CXPI bus signal searching in binary notation.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DPATtern:
PATtern {<String>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DPATtern:PATtern?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:
PATTERN "00110101"
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:PATTERN?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:
PATTERN "00110101"

Description The number of bytes that can be set varies depending on the :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:DBYTE setting.

5.24 SERIALbus group

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:ENDian

Function Sets or queries the byte order of the reference values, which is one of the ID and data conditions, for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:ENDian {BIG|LITtle}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:ENDian?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:ENDIAN?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:MODE

Function Sets or queries the ID and data condition (enabled/disabled) for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:MODE {<Boolean>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:MODE?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:MODE ON
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:MODE?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:MODE 1

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:MSBLsb

Function Sets or queries the MSB and LSB bits for the data, which is one of the ID and data conditions, for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:MSBLsb {<NRf>,<NRf>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:MSBLsb?
<x> = 1 to 4
<NRf> = 0 to 63

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 7,0
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:MSBLSB?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 7,0

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:SIGN

Function Sets or queries the sign, which is one of the ID and data conditions, for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:SIGN {SIGN|UNSign}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:DATA:SIGN?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:SIGN?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation?

Function Queries all frame information settings for the ID and data conditions for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation?
<x> = 1 to 4

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:CT

Function Sets or queries the frame information counter value, which is one of the ID and data conditions, for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:CT {<NRf>|DONTcare}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:CT?
<x> = 1 to 4
<NRf> = 0 to 3

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:FINFORMATION:CT 3
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:FINFORMATION:CT?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:FINFORMATION:CT 3

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:MODE

Function Sets or queries the frame information (enabled/disabled), which is one of the ID and data conditions, for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:MODE {<Boolean>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:MODE?

Example :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:MODE ON
:SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:MODE?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:MODE 1

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:SLEEP

Function Sets or queries the frame information sleep bit, which is one of the ID and data conditions, for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:SLEEP {0|1|X}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:SLEEP?

Example :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:SLEEP 1
:SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:SLEEP?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:SLEEP 1

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:WAKEup

Function Sets or queries the frame information wakeup bit, which is one of the ID and data conditions, for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:WAKEup {0|1|X}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:FINformation:WAKEup?

Example :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:WAKEUP 1
:SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:WAKEUP?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:FINFORMATION:WAKEUP 1

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID?

Function Queries all ID settings for the ID and data conditions for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID?
<x> = 1 to 4

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:HEXa

Function Sets an ID, which is one of the ID and data conditions, for CXPI bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:ID:HEXa "1E"

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:MODE

Function Sets or queries the ID condition (enabled/disabled), which is one of the ID and data conditions, for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:MODE {<Boolean>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:MODE?

Example :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:ID:MODE ON
:SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:ID:MODE?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:ID:MODE 1

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:PATtern

Function Sets or queries the ID, which is one of the ID and data conditions, for CXPI bus signal searching in binary notation.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:PATtern {<String>}
:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:PATtern?

Example :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:ID:PATTERN "0010000"
:SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:ID:PATTERN?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:IDDATA:ID:PATTERN "0010000"

5.24 SERIALbus group

:SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:PTYPE

Function Sets or queries the PTYPE comparison condition, which is one of the ID and data conditions, for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup[:IDData]:ID:PTYPE {DONTcare|NO|YES}
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:ID:PTYPE DONTCARE
:SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:ID:PTYPE?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
IDDATA:ID:PTYPE DONTCARE

:SERIALbus<x>:CXPI:SEARCH:SETup:MODE

Function Sets or queries the CXPI bus signal search type.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
MODE {ERRor|IDData|PTYPE|SOF|
WAKEupsleep}
:SERIALbus<x>:CXPI:SEARCH:SETup:
MODE?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
MODE SOF
:SERIALBUS1:CXPI:SEARCH:SETUP:MODE?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
MODE SOF

:SERIALbus<x>:CXPI:SEARCH:SETup:PTYPE

Function Queries all CXPI bus signal search PTYPE settings.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
PTYPE?
<x> = 1 to 4

:SERIALbus<x>:CXPI:SEARCH:SETup:PTYPE:CONDition

Function Sets or queries the CXPI bus signal search PTYPE condition setting.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
PTYPE:CONDition {EPTYPE|NOResponse}
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:PTYPE:
CONDITION EPTYPE
:SERIALBUS1:CXPI:SEARCH:SETUP:PTYPE:
CONDITION?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
PTYPE:CONDITION EPTYPE

Description For EPTYPE, all PTYPEs are searched for. For NOResponse, PTYPEs without responses are searched for.

:SERIALbus<x>:CXPI:SEARCH:SETup:WAKEup sleep?

Function Queries all wakeup and sleep settings for CXPI bus signal searching.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
WAKEupsleep?
<x> = 1 to 4

:SERIALbus<x>:CXPI:SEARCH:SETup:WAKEup sleep:SFRame

Function Sets or queries the CXPI bus signal search sleep frame setting.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
WAKEupsleep:SFRame {<Boolean>}
:SERIALbus<x>:CXPI:SEARCH:SETup:
WAKEupsleep:SFRame?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
WAKEUPSLEEP:SFRAME ON
:SERIALBUS1:CXPI:SEARCH:SETUP:
WAKEUPSLEEP:SFRAME?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
WAKEUPSLEEP:SFRAME 1

:SERIALbus<x>:CXPI:SEARCH:SETup:WAKEup sleep:SLEEP

Function Sets or queries the CXPI bus signal search sleep (clock unavailable condition) setting.

Syntax :SERIALbus<x>:CXPI:SEARCH:SETup:
WAKEupsleep:SLEEP {<Boolean>}
:SERIALbus<x>:CXPI:SEARCH:SETup:
WAKEupsleep:SLEEP?
<x> = 1 to 4

Example :SERIALBUS1:CXPI:SEARCH:SETUP:
WAKEUPSLEEP:SLEEP ON
:SERIALBUS1:CXPI:SEARCH:SETUP:
WAKEUPSLEEP:SLEEP?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:
WAKEUPSLEEP:SLEEP 1

:SERIALbus<x>:CXPI:SEARCh:SEtUp:WAKEup sleep:WAKEup

Function Sets or queries the CXPI bus signal search wakeup (clock available condition) setting.

Syntax :SERIALbus<x>:CXPI:SEARCh:SEtUp:WAKEupsleep:WAKEup {<Boolean>}
:SERIALbus<x>:CXPI:SEARCh:SEtUp:WAKEupsleep:WAKEup?

Example :SERIALBUS1:CXPI:SEARCH:SETUP:WAKEUPSLEEP:WAKEUP ON
:SERIALBUS1:CXPI:SEARCH:SETUP:WAKEUPSLEEP:WAKEUP?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:WAKEUPSLEEP:WAKEUP 1

:SERIALbus<x>:CXPI:SEARCh:SEtUp:WAKEup sleep:WPULse

Function Sets or queries the CXPI bus signal search wakeup pulse setting.

Syntax :SERIALbus<x>:CXPI:SEARCh:SEtUp:WAKEupsleep:WPULse {<Boolean>}
:SERIALbus<x>:CXPI:SEARCh:SEtUp:WAKEupsleep:WPULse?

Example :SERIALBUS1:CXPI:SEARCH:SETUP:WAKEUPSLEEP:WPULSE ON
:SERIALBUS1:CXPI:SEARCH:SETUP:WAKEUPSLEEP:WPULSE?
-> :SERIALBUS1:CXPI:SEARCH:SETUP:WAKEUPSLEEP:WPULSE 1

:SERIALbus<x>:DECode?

Function Queries all settings related to the decoding of serial bus signals.

Syntax :SERIALbus<x>:DECode?
<x> = 1 to 4

:SERIALbus<x>:DECode[:FORMat]

Function Sets or queries the decode display format of serial bus signals.

Syntax :SERIALbus<x>:DECode[:FORMat] {BINARY|DECIMAL|HEXa|AScii|SYMBOL}
:SERIALbus<x>:DECode[:FORMat]?

Example :SERIALBUS1:DECODE:FORMAT BINARY
:SERIALBUS1:DECODE:FORMAT?
-> :SERIALBUS1:DECODE:FORMAT BINARY

:SERIALbus<x>:DECode:LABel

Function Sets or queries the decode label name of serial bus signals.

Syntax :SERIALbus<x>:DECode:LABel {<String>}
:SERIALbus<x>:DECode:LABel?
<x> = 1 to 4
<String> = Up to 8 characters

Example :SERIALBUS1:DECODE:LABEL "S1"
:SERIALBUS1:DECODE:LABEL?
-> :SERIALBUS1:DECODE:LABEL "S1"

Description You cannot use this command or query if :SERIALbus<x>:TYPE is set to SPI or SENT.

:SERIALbus<x>:DECode:POSition

Function Sets or queries the vertical position for decoding serial bus signals.

Syntax :SERIALbus<x>:DECode:POSition {AUTO|MANual,<NRF>}
:SERIALbus<x>:DECode:POSition?
<x> = 1 to 4
MANual <NRF> = -3.8 to 3.8 (div), (10 div/display record length) steps

Example :SERIALBUS1:DECODE:POSITION AUTO
:SERIALBUS1:DECODE:POSITION?
-> :SERIALBUS1:DECODE:POSITION AUTO

Description You cannot use this command or query if :SERIALbus<x>:TYPE is set to SPI or SENT.

:SERIALbus<x>:DECode:SENT?

Function Queries all SENT signal decode settings.

Syntax :SERIALbus<x>:DECode:SENT?
<x> = 1 to 4

:SERIALbus<x>:DECode:SENT:FAST?

Function Queries all settings for decoding the fast channel of SENT signals.

Syntax :SERIALbus<x>:DECode:SENT:FAST?
<x> = 1 to 4

:SERIALbus<x>:DECode:SENT:FAST:LABel

Function Sets or queries the label name for decoding the fast channel of SENT signals.

Syntax :SERIALbus<x>:DECode:SENT:FAST:LABel {<String>}
:SERIALbus<x>:DECode:SENT:FAST:LABel?
<x> = 1 to 4
<String> = Up to 8 characters

Example :SERIALBUS1:DECODE:SENT:FAST:LABel "S1-F"
:SERIALBUS1:DECODE:SENT:FAST:LABel?
-> :SERIALBUS1:DECODE:SENT:FAST:LABel "S1-F"

5.24 SERIALbus group

:SERIALbus<x>:DECODE:SENT:FAST:POSITION

Function Sets or queries the vertical position for decoding the fast channel of SENT signals.

Syntax :SERIALbus<x>:DECODE:SENT:FAST:
POSITION {AUTO|MANual,<NRf>}
:SERIALbus<x>:DECODE:SENT:FAST:
POSITION?
<x> = 1 to 4
MANual <NRf> = -3.8 to 3.8 (div), 10 div/display
record length steps

Example :SERIALBUS1:DECODE:SENT:FAST:
POSITION AUTO
:SERIALBUS1:DECODE:SENT:FAST:
POSITION?
-> :SERIALBUS1:DECODE:SENT:FAST:
POSITION AUTO

:SERIALbus<x>:DECODE:SENT:SLOW?

Function Queries all settings for decoding the slow channel of SENT signals.

Syntax :SERIALbus<x>:DECODE:SENT:SLOW?
<x> = 1 to 4

:SERIALbus<x>:DECODE:SENT:SLOW:LABEL

Function Sets or queries the label name for decoding the slow channel of SENT signals.

Syntax :SERIALbus<x>:DECODE:SENT:SLOW:
LABEL {<String>}
:SERIALbus<x>:DECODE:SENT:SLOW:
LABEL?
<x> = 1 to 4
<String> = Up to 8 characters

Example :SERIALBUS1:DECODE:SENT:SLOW:
LABEL "S1-S"
:SERIALBUS1:DECODE:SENT:SLOW:LABEL?
-> :SERIALBUS1:DECODE:SENT:SLOW:
LABEL "S1-S"

:SERIALbus<x>:DECODE:SENT:SLOW:POSITION

Function Sets or queries the vertical position for decoding the slow channel of SENT signals.

Syntax :SERIALbus<x>:DECODE:SENT:SLOW:
POSITION {AUTO|MANual,<NRf>}
:SERIALbus<x>:DECODE:SENT:SLOW:
POSITION?
<x> = 1 to 4
MANual <NRf> = -3.8 to 3.8 (div), 10 div/display
record length steps

Example :SERIALBUS1:DECODE:SENT:SLOW:
POSITION AUTO
:SERIALBUS1:DECODE:SENT:SLOW:
POSITION?
-> :SERIALBUS1:DECODE:SENT:SLOW:
POSITION AUTO

:SERIALbus<x>:DECODE:SPI?

Function Queries all SPI bus signal decode settings.

Syntax :SERIALbus<x>:DECODE:SPI?
<x> = 1 to 4
<y> = 1 to 2

:SERIALbus<x>:DECODE:SPI:DATA<y>?

Function Queries all settings related to the specified data for SPI bus signal decoding.

Syntax :SERIALbus<x>:DECODE:SPI:DATA<y>?
<x> = 1 to 4
<y> = 1 to 2

:SERIALbus<x>:DECODE:SPI:DATA<y>:LABEL

Function Sets or queries the label name for decoding SPI bus signals.

Syntax :SERIALbus<x>:DECODE:SPI:DATA<y>:
LABEL {<String>}
:SERIALbus<x>:DECODE:SPI:DATA<y>:
LABEL?
<x> = 1 to 4
<y> = 1 to 2
<String> = Up to 8 characters

Example :SERIALBUS1:DECODE:SPI:DATA1:
LABEL "S1-1"
:SERIALBUS1:DECODE:SPI:DATA1:LABEL?
-> :SERIALBUS1:DECODE:SPI:DATA1:
LABEL "S1-1"

:SERIALbus<x>:DECODE:SPI:DATA<y>:POSITION

Function Sets or queries the vertical position for decoding SPI bus signals.

Syntax :SERIALbus<x>:DECODE:SPI:DATA<y>:
POSITION {AUTO|MANual,<Nrf>}
:SERIALbus<x>:DECODE:SPI:DATA<y>:
POSITION?
<x> = 1 to 4
<y> = 1 to 2
MANual <Nrf> = -3.8 to 3.8 (div), 10 div/display
record length steps

Example :SERIALBUS1:DECODE:SPI:DATA1:
POSITION AUTO
:SERIALBUS1:DECODE:SPI:DATA1:
POSITION?
-> :SERIALBUS1:DECODE:SPI:DATA1:
POSITION AUTO

:SERIALbus<x>:DISPLAY

Function Sets or queries whether serial bus signal analysis is to be performed.

Syntax :SERIALbus<x>:DISPLAY {<Boolean>}
:SERIALbus<x>:DISPLAY?
<x> = 1 to 4

Example :SERIALBUS1:DISPLAY ON
:SERIALBUS1:DISPLAY?
-> :SERIALBUS1:DISPLAY 1

:SERIALbus<x>:FLEXray?

Function Queries all FlexRay bus signal analysis and search settings.

Syntax :SERIALbus<x>:FLEXray?
<x> = 1 to 4

:SERIALbus<x>:FLEXray:ANALyze?

Function Queries all FlexRay bus signal analysis settings.

Syntax :SERIALbus<x>:FLEXray:ANALyze?
<x> = 1 to 4

:SERIALbus<x>:FLEXray[:ANALyze]:SETup?

Function Queries all FlexRay bus signal analysis bus settings.

Syntax :SERIALbus<x>:FLEXray[:ANALyze]:
SETup?
<x> = 1 to 4

:SERIALbus<x>:FLEXray[:ANALyze]:SETup: BChannel

Function Sets or queries the channel bus type for FlexRay bus signal analysis.

Syntax :SERIALbus<x>:FLEXray[:ANALyze]:
SETup:BChannel {A|B}
:SERIALbus<x>:FLEXray[:ANALyze]:
SETup:BChannel?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:ANALYZE:SETUP:
BCHANNEL A
:SERIALBUS1:FLEXRAY:ANALYZE:SETUP:
BCHANNEL?
-> :SERIALBUS1:FLEXRAY:ANALYZE:
SETUP:BCHANNEL A

:SERIALbus<x>:FLEXray[:ANALyze]:SETup: BRATe

Function Sets or queries the FlexRay bus signal analysis bit rate (data transfer rate).

Syntax :SERIALbus<x>:FLEXray[:ANALyze]:
SETup:BRATe {<Nrf>}
:SERIALbus<x>:FLEXray[:ANALyze]:
SETup:BRATe?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:ANALYZE:SETUP:
BRATE 5000000
:SERIALBUS1:FLEXRAY:ANALYZE:SETUP:
BRATE?
-> :SERIALBUS1:FLEXRAY:ANALYZE:
SETUP:BRATE 5000000

:SERIALbus<x>:FLEXray[:ANALyze]:SETup: SOURce

Function Sets or queries the FlexRay bus signal analysis source.

Syntax :SERIALbus<x>:FLEXray[:ANALyze]:
SETup:SOURce {<Nrf>|MATH<y>}
:SERIALbus<x>:FLEXray[:ANALyze]:
SETup:SOURce?
<x> = 1 to 4
<Nrf> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:FLEXRAY:ANALYZE:SETUP:
SOURCE 1
:SERIALBUS1:FLEXRAY:ANALYZE:SETUP:
SOURCE?
-> :SERIALBUS1:FLEXRAY:ANALYZE:
SETUP:SOURCE 1

5.24 SERIALbus group

:SERIALbus<x>:FLEXray:DETail?

Function Queries all FlexRay bus signal analysis result list settings.

Syntax :SERIALbus<x>:FLEXray:DETail?
<x> = 1 to 4

:SERIALbus<x>:FLEXray:DETail:DISPlay

Function Sets or queries the display mode for the FlexRay bus signal analysis result list.

Syntax :SERIALbus<x>:FLEXray:DETail:
DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:FLEXray:DETail:
DISPlay?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:DETAIL:
DISPLAY FULL
-> :SERIALBUS1:FLEXRAY:DETAIL:DISPLAY?
-> :SERIALBUS1:FLEXRAY:DETAIL:
DISPLAY FULL

:SERIALbus<x>:FLEXray:DETail:LIST:ITEM?

Function Queries all items that will be displayed in the FlexRay bus signal analysis result list.

Syntax :SERIALbus<x>:FLEXray:DETail:LIST:
ITEM?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:DETAIL:LIST:
ITEM?
-> :SERIALBUS1:FLEXRAY:DETAIL:LIST:
ITEM "No.,Time(ms),S/D,IND,ID,Len,
CC,Data,Information,"

:SERIALbus<x>:FLEXray:DETail:LIST:VALUE?

Function Queries all the data for the specified analysis number in the FlexRay bus signal analysis result list.

Syntax :SERIALbus<x>:FLEXray:DETail:LIST:
VALue? {<NRf>|MAXimum|MINimum}
<x> = 1 to 4
<NRf> = See the Features Guide for this information.

Example :SERIALBUS1:FLEXRAY:DETAIL:LIST:
VALUE? 0
-> :SERIALBUS1:FLEXRAY:DETAIL:LIST:
VALUE "0,-0.0001968,S,1111, 4, 4,2,
01 02 03 04 05 06 07 08,,,"

:SERIALbus<x>:FLEXray:SEARCh?

Function Queries all FlexRay bus signal search settings.

Syntax :SERIALbus<x>:FLEXray:SEARCh?
<x> = 1 to 4

:SERIALbus<x>:FLEXray:SEARCh:ABORt

Function Aborts the FlexRay bus signal search.

Syntax :SERIALbus<x>:FLEXray:SEARCh:ABORt
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:ABORT

:SERIALbus<x>:FLEXray:SEARCh:EXECute

Function Executes a FlexRay bus signal search.

Syntax :SERIALbus<x>:FLEXray:SEARCh:EXECute
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:EXECUTE

:SERIALbus<x>:FLEXray:SEARCh:FJUMp:CCOUNT

Function Jumps to the cycle count field in the FlexRay bus signal search result.

Syntax :SERIALbus<x>:FLEXray:SEARCh:FJUMp:
CCount
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:FJUMP:
CCOUNT

:SERIALbus<x>:FLEXray:SEARCh:FJUMp:CRC

Function Jumps to the CRC field in the FlexRay bus signal search result.

Syntax :SERIALbus<x>:FLEXray:SEARCh:FJUMp:
CRC
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:FJUMP:CRC

:SERIALbus<x>:FLEXray:SEARCh:FJUMp:FRAMEid

Function Jumps to the frame ID field in the FlexRay bus signal search result.

Syntax :SERIALbus<x>:FLEXray:SEARCh:FJUMp:
FRAMEid
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:FJUMP:
FRAMEID

:SERIALbus<x>:FLEXray:SEARCh:FJUMp:HCRC

Function Jumps to the header CRC field in the FlexRay bus signal search result.

Syntax :SERIALbus<x>:FLEXray:SEARCh:FJUMp:
HCRC
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:FJUMP:
HCRC

:SERIALbus<x>:FLEXray:SEARCH:FJUMP:PLENGTH

Function Jumps to the payload length field in the FlexRay bus signal search result.

Syntax :SERIALbus<x>:FLEXray:SEARCH:FJUMP:PLENGTH
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:FJUMP:PLENGTH

:SERIALbus<x>:FLEXray:SEARCH:SELEct

Function Sets the detected waveform number for a FlexRay bus signal search and queries the zoom position corresponding to that number.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SELEct {<Nrf>|MAXimum}
:SERIALbus<x>:FLEXray:SEARCH:SELEct?<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SELECT 1
:SERIALBUS1:FLEXRAY:SEARCH:SELECT?
-> :SERIALBUS1:FLEXRAY:SEARCH:SELECT 1.50000000

:SERIALbus<x>:FLEXray:SEARCH:SETup?

Function Queries all FlexRay bus signal search condition settings.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETup?<x> = 1 to 4

:SERIALbus<x>:FLEXray:SEARCH:SETup:ERor?

Function Queries all FlexRay bus signal search error settings.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETup:ERor?<x> = 1 to 4

:SERIALbus<x>:FLEXray:SEARCH:SETup:ERor:BSS

Function Sets or queries the FlexRay bus signal search BSS Error setting.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETup:ERor:BSS {<Boolean>}
:SERIALbus<x>:FLEXray:SEARCH:SETup:ERor:BSS?<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:BSS ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:BSS?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:BSS 1

:SERIALbus<x>:FLEXray:SEARCH:SETup:ERor:CRC

Function Sets or queries the FlexRay bus signal search CRC Error setting.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETup:ERor:CRC {<Boolean>}
:SERIALbus<x>:FLEXray:SEARCH:SETup:ERor:CRC?<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:CRC ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:CRC?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:CRC 1

:SERIALbus<x>:FLEXray:SEARCH:SETup:ERor:FES

Function Sets or queries the FlexRay bus signal search FES Error setting.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETup:ERor:FES {<Boolean>}
:SERIALbus<x>:FLEXray:SEARCH:SETup:ERor:FES?<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:FES ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:FES?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:FES 1

:SERIALbus<x>:FLEXray:SEARCH:SETup:ERor:HCRC

Function Sets or queries the FlexRay bus signal search Header CRC Error setting.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETup:ERor:HCRC {<Boolean>}
:SERIALbus<x>:FLEXray:SEARCH:SETup:ERor:HCRC?<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:HCRC ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:HCRC?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:ERROR:HCRC 1

5.24 SERIALbus group

:SERIALbus<x>:FLEXray:SEARCh:SETup:IDData?

Function Queries all ID and data condition settings for FlexRay bus signal search.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup:IDData?
<x> = 1 to 4

:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT?

Function Queries all FlexRay bus signal search cycle count settings.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT?
<x> = 1 to 4

:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:CONDition

Function Sets or queries the FlexRay bus signal search cycle count data condition.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:CONDition {EQUAL|GREATER|INSIDE|LESS|NOTEQUAL|OUTSIDE}
:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:CONDITION EQUAL
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:CONDITION?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:CONDITION EQUAL

:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:COUNT<y>

Function Sets or queries the FlexRay bus signal search cycle count setting.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:COUNT<y> {<NRF>}
:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:COUNT<y>?
<x> = 1 to 4
<y> = 1 to 2
<NRF> = 0 to 63

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:COUNT1 10
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:COUNT1?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:COUNT1 10

Description

- **When :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:CONDitionEQUAL|GREATER|NOTEQUAL is specified, use :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:COUNT1 to set the value.**
- **When :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:CONDitionLESS is specified, use :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:COUNT2 to set the value.**
- **When :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:CONDitionBETWEEN|NOTBetween is specified, use :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:COUNT1 to set the smaller value and :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:COUNT2 to set the larger value.**

:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:MODE

Function Sets or queries the cycle count mode, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:MODE {<Boolean>}
:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:CCOUNT:MODE?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:MODE ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:MODE?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:CCOUNT:MODE 1

:SERIALbus<x>:FLEXray:SEARCh:SETup[:ID Data]:DATA?

Function Queries all data settings for the ID and data conditions for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:DATA?
<x> = 1 to 4

:SERIALbus<x>:FLEXray:SEARCh:SETup[:ID Data]:DATA:BCount

Function Sets or queries the pattern comparison position, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:DATA:BCount {<Nrf>}
:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:DATA:BCount?
<x> = 1 to 4
<Nrf> = 0 to 253

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:BCOUNT 1
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:BCOUNT?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:BCOUNT 1

:SERIALbus<x>:FLEXray:SEARCh:SETup[:ID Data]:DATA:CONDition

Function Sets or queries the data comparison condition, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:DATA:CONDition {EQUAL|FALSE|GREATER|INSIDE|LESS|NOTEQUAL|OUTSIDE|TRUE}
:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:DATA:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:CONDITION EQUAL
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:CONDITION?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:CONDITION EQUAL

:SERIALbus<x>:FLEXray:SEARCh:SETup[:ID Data]:DATA:DBYTE

Function Sets or queries the number of data bytes, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:DATA:DBYTE {<Nrf>}
:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:DATA:DBYTE?
<x> = 1 to 4
<Nrf> = 1 to 8

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DBYTE 1
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DBYTE?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DBYTE 1

:SERIALbus<x>:FLEXray:SEARCh:SETup[:ID Data]:DATA:DECimal<y>

Function Sets or queries a reference value, which is one of the ID and data conditions, for FlexRay bus signal searching in decimal notation.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:DATA:DECimal<y> {<Nrf>}
:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:DATA:DECimal<y>?
<x> = 1 to 4
<y> = 1 to 2
<Nrf> = See the Features Guide for this information.

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1 1
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1

:SERIALbus<x>:FLEXray:SEARCh:SETup[:ID Data]:DATA:DPATtern?

Function Queries all judgment data settings for the ID and data conditions for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:DATA:DPATtern?
<x> = 1 to 4

5.24 SERIALbus group

:SERIALbus<x>:FLEXray:SEARCH:SETUP[:ID Data]:DATA:DPATtern:HEXa

Function Sets a reference value, which is one of the ID and data conditions, for FlexRay bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETUP[:IDData]:DATA:DPATtern:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DPATtern:HEXA "12"

Description The number of bytes that can be set varies depending on the :SERIALbus<x>:FLEXray:SEARCH:SETUP[:IDData]:DATA:DBYTE setting.

:SERIALbus<x>:FLEXray:SEARCH:SETUP[:ID Data]:DATA:DPATtern:PATtern

Function Sets or queries the data, which is one of the ID and data conditions, for FlexRay bus signal searching in binary notation.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETUP[:IDData]:DATA:DPATtern:
PATtern {<String>}
:SERIALbus<x>:FLEXray:SEARCH:SETUP[:IDData]:DATA:DPATtern:PATtern?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DPATtern:
PATTERN "00110101"
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DPATtern:PATTERN?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:DPATtern:
PATTERN "00110101"

Description The number of bytes that can be set varies depending on the :SERIALbus<x>:FLEXray:SEARCH:SETUP[:IDData]:DATA:DBYTE setting.

:SERIALbus<x>:FLEXray:SEARCH:SETUP[:ID Data]:DATA:ENDian

Function Sets or queries the byte order of the reference values, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETUP[:IDData]:DATA:ENDian {BIG|LITtle}
:SERIALbus<x>:FLEXray:SEARCH:SETUP[:IDData]:DATA:ENDian?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:ENDIAN?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG

:SERIALbus<x>:FLEXray:SEARCH:SETUP[:ID Data]:DATA:MODE

Function Sets or queries the data condition (enabled/disabled), which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETUP[:IDData]:DATA:MODE {<Boolean>}
:SERIALbus<x>:FLEXray:SEARCH:SETUP[:IDData]:DATA:MODE?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:MODE ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:MODE?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:MODE 1

:SERIALbus<x>:FLEXray:SEARCH:SETUP[:ID Data]:DATA:MSBLsb

Function Sets or queries the MSB and LSB bits for the data, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETUP[:IDData]:DATA:MSBLsb {<Nrf>,<Nrf>}
:SERIALbus<x>:FLEXray:SEARCH:SETUP[:IDData]:DATA:MSBLsb?
<x> = 1 to 4
<Nrf> = 0 to 63

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 7,0
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:MSBLSB?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 7,0

:SERIALbus<x>:FLEXray:SEARCH:SETUP[:ID Data]:DATA:SIGN

Function Sets or queries the sign, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETUP[:IDData]:DATA:SIGN {SIGN|UNSign}
:SERIALbus<x>:FLEXray:SEARCH:SETUP[:IDData]:DATA:SIGN?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:SIGN?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN

:SERIALbus<x>:FLEXray:SEARCh:SETup[:ID Data]:FID?

Function Queries the frame ID, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:FID?
<x> = 1 to 4

:SERIALbus<x>:FLEXray:SEARCh:SETup[:ID Data]:FID:CONDition

Function Sets or queries the frame ID data condition, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:FID:CONDition {EQUAL|GREATER|INSIDE|LESS|NOTEQUAL|OUTSIDE}
:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:FID:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:CONDITION EQUAL
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:CONDITION?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:CONDITION EQUAL

:SERIALbus<x>:FLEXray:SEARCh:SETup[:ID Data]:FID:ID<y>

Function Sets or queries the frame ID value, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:FID:ID<y> {<NRF>}
:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:FID:ID<y>?
<x> = 1 to 4
<y> = 1, 2
<NRF> = 1 to 2047

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:ID1 100
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:ID1?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:ID1 100

:SERIALbus<x>:FLEXray:SEARCh:SETup[:ID Data]:FID:MODE

Function Sets or queries the frame ID condition (enabled/disabled), which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:FID:MODE {<Boolean>}
:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:FID:MODE?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:MODE ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:MODE?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:FID:MODE 1

:SERIALbus<x>:FLEXray:SEARCh:SETup[:ID Data]:INDicator?

Function Queries the indicator, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:INDicator?
<x> = 1 to 4

:SERIALbus<x>:FLEXray:SEARCh:SETup[:ID Data]:INDicator:MODE

Function Sets or queries the indicator condition (enabled/disabled), which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:INDicator:MODE {<Boolean>}
:SERIALbus<x>:FLEXray:SEARCh:SETup[:IDData]:INDicator:MODE?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:MODE ON
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:MODE?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:MODE 1

5.24 SERIALbus group

:SERIALbus<x>:FLEXray:SEARCH:SETup[:ID Data]:INDicator:NFRame

Function Sets or queries the indicator null frame, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:INDicator:NFRame {0|1|X}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:INDicator:NFRame?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:NFRAME 1
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:NFRAME?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:NFRAME 1

:SERIALbus<x>:FLEXray:SEARCH:SETup[:ID Data]:INDicator:PPReamble

Function Sets or queries the indicator payload preamble, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:INDicator:PPReamble {0|1|X}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:INDicator:PPReamble?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:PPREMBLE 1
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:PPREMBLE?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:PPREMBLE 1

:SERIALbus<x>:FLEXray:SEARCH:SETup[:ID Data]:INDicator:STFrame

Function Sets or queries the indicator start frame, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:INDicator:STFrame {0|1|X}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:INDicator:STFrame?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:STFRAME 1
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:STFRAME?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:STFRAME 1

:SERIALbus<x>:FLEXray:SEARCH:SETup[:ID Data]:INDicator:SYFrame

Function Sets or queries the indicator sync frame, which is one of the ID and data conditions, for FlexRay bus signal searching.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:INDicator:SYFrame {0|1|X}
:SERIALbus<x>:FLEXray:SEARCH:SETup[:IDData]:INDicator:SYFrame?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:SYFRAME 1
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:SYFRAME?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
IDDATA:INDICATOR:SYFRAME 1

:SERIALbus<x>:FLEXray:SEARCH:SETup:MODE

Function Sets or queries the FlexRay bus signal search mode setting.

Syntax :SERIALbus<x>:FLEXray:SEARCH:SETup:MODE {ERROR|FStart|IDData}
:SERIALbus<x>:FLEXray:SEARCH:SETup:MODE?
<x> = 1 to 4

Example :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
MODE FSTART
:SERIALBUS1:FLEXRAY:SEARCH:SETUP:
MODE?
-> :SERIALBUS1:FLEXRAY:SEARCH:SETUP:
MODE FSTART

:SERIALbus<x>:I2C?

Function Queries all I2C bus signal analysis and search settings.

Syntax :SERIALbus<x>:I2C?
<x> = 1 to 4

:SERIALbus<x>:I2C:ANALyze?

Function Queries all I2C bus signal analysis settings.

Syntax :SERIALbus<x>:I2C:ANALyze?
<x> = 1 to 4

:SERIALbus<x>:I2C[:ANALyze]:SETup?

Function Queries all I2C bus signal analysis bus settings.

Syntax :SERIALbus<x>:I2C[:ANALyze]:SETup?
<x> = 1 to 4

:SERIALbus<x>:I2C[:ANALyze]:SETup:CLOCK?

Function Queries all I2C bus signal analysis clock settings.

Syntax :SERIALbus<x>:I2C[:ANALyze]:SETup:
CLOCK?
<x> = 1 to 4

:SERIALbus<x>:I2C[:ANALyze]:SETup:CLOCK:SOURce

Function Sets or queries the I2C bus signal analysis clock.

Syntax :SERIALbus<x>:I2C[:ANALyze]:SETup:
CLOCK:SOURce {<NRf>|BIT<y>|MATH<y>}
:SERIALbus<x>:I2C[:ANALyze]:SETup:
CLOCK:SOURce?
<x> = 1 to 4
<NRf> = 1 to 4
BIT<y>'s <y> = 1 to 8
MATH<y>'s <y> = 1 to 4

Example :SERIALBUS1:I2C:ANALYZE:SETUP:CLOCK:
SOURCE 1
:SERIALBUS1:I2C:ANALYZE:SETUP:CLOCK:
SOURCE?
-> :SERIALBUS1:I2C:ANALYZE:SETUP:
CLOCK:SOURCE 1

:SERIALbus<x>:I2C[:ANALyze]:SETup:DATA?

Function Queries all I2C bus signal analysis data settings.

Syntax :SERIALbus<x>:I2C[:ANALyze]:SETup:
DATA?
<x> = 1 to 4

:SERIALbus<x>:I2C[:ANALyze]:SETup:DATA:SOURce

Function Sets or queries the I2C bus signal analysis data.

Syntax :SERIALbus<x>:I2C[:ANALyze]:SETup:
DATA:SOURce {<NRf>|BIT<y>|MATH<y>}
:SERIALbus<x>:I2C[:ANALyze]:SETup:
DATA:SOURce?
<x> = 1 to 4
<NRf> = 1 to 4
BIT<y>'s <y> = 1 to 8
MATH<y>'s <y> = 1 to 4

Example :SERIALBUS1:I2C:ANALYZE:SETUP:DATA:
SOURCE 1
:SERIALBUS1:I2C:ANALYZE:SETUP:DATA:
SOURCE?
-> :SERIALBUS1:I2C:ANALYZE:SETUP:
DATA:SOURCE 1

:SERIALbus<x>:I2C[:ANALyze]:SETup:INCLuderw

Function Sets or queries the enable/disable condition of address R/W for I2C bus signal analysis.

Syntax :SERIALbus<x>:I2C[:ANALyze]:SETup:
INCLuderw {<Boolean>}
:SERIALbus<x>:I2C[:ANALyze]:SETup:
INCLuderw?
<x> = 1 to 4

Example :SERIALBUS1:I2C:ANALYZE:SETUP:
INCLUDERW ON
:SERIALBUS1:I2C:ANALYZE:SETUP:
INCLUDERW?
-> :SERIALBUS1:I2C:ANALYZE:SETUP:
INCLUDERW 1

:SERIALbus<x>:I2C:DETAil?

Function Queries all I2C bus signal analysis result list settings.

Syntax :SERIALbus<x>:I2C:DETAil?
<x> = 1 to 4

:SERIALbus<x>:I2C:DETAil:DISPlay

Function Sets or queries the display mode for the I2C bus signal analysis result list.

Syntax :SERIALbus<x>:I2C:DETAil:
DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:I2C:DETAil:DISPlay?
<x> = 1 to 4

Example :SERIALBUS1:I2C:DETAIL:DISPLAY FULL
:SERIALBUS1:I2C:DETAIL:DISPLAY?
-> :SERIALBUS1:I2C:DETAIL:
DISPLAY FULL

:SERIALbus<x>:I2C:DETAil:LIST:ITEM?

Function Queries all items that will be displayed in the I2C bus signal analysis result list.

Syntax :SERIALbus<x>:I2C:DETAil:LIST:ITEM?
<x> = 1 to 4

Example :SERIALBUS1:I2C:DETAIL:LIST:ITEM?
-> :SERIALBUS1:I2C:DETAIL:LIST:
ITEM "No.,Time(ms),1st,2nd,R/W,Data,
Information,"

5.24 SERIALbus group

:SERIALbus<x>:I2C:DETAIL:LIST:VALUE?

Function Queries all the data for the specified analysis number in the I2C bus signal analysis result list.

Syntax :SERIALbus<x>:I2C:DETAIL:LIST:
VALUE? {<Nrf>|MAXimum|MINimum}
<x> = 1 to 4
<Nrf> = See the Features Guide for this information.

Example :SERIALBUS1:I2C:DETAIL:LIST:VALUE? 0
-> :SERIALBUS1:I2C:DETAIL:LIST:
VALUE "0,-0.07000,AA*,W,AE*88*,
7-bit,"

:SERIALbus<x>:I2C:SEARCH?

Function Queries all I2C bus signal search settings.

Syntax :SERIALbus<x>:I2C:SEARCH?
<x> = 1 to 4

:SERIALbus<x>:I2C:SEARCH:ABORT

Function Aborts the I2C bus signal search.

Syntax :SERIALbus<x>:I2C:SEARCH:ABORT
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:ABORT

:SERIALbus<x>:I2C:SEARCH:EXECute

Function Executes an I2C bus signal search.

Syntax :SERIALbus<x>:I2C:SEARCH:EXECute
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:EXECUTE

:SERIALbus<x>:I2C:SEARCH:SElect

Function Sets which detected point to display in the I2C bus signal search zoom window and queries the zoom position of the detected point.

Syntax :SERIALbus<x>:I2C:SEARCH:
SElect {<Nrf>|MAXimum}
:SERIALbus<x>:I2C:SEARCH:SElect?
<x> = 1 to 4
<Nrf> = 0 to 49999

Example :SERIALBUS1:I2C:SEARCH:SELECT 1
:SERIALBUS1:I2C:SEARCH:SELECT?
-> :SERIALBUS1:I2C:SEARCH:
SELECT 1.50000000

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:I2C:SEARCH:SETup?

Function Queries all I2C bus signal search condition settings.

Syntax :SERIALbus<x>:I2C:SEARCH:SETup?
<x> = 1 to 4

:SERIALbus<x>:I2C:SEARCH[:SETup]:ADDRESS?

Function Queries all I2C bus signal search address pattern settings.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETup]:
ADDRESS?
<x> = 1 to 4

:SERIALbus<x>:I2C:SEARCH[:SETup]:ADDRESS:ADDRESS?

Function Queries all settings related to the address of the I2C bus signal search address pattern.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETup]:
ADDRESS:ADDRESS?
<x> = 1 to 4

:SERIALbus<x>:I2C:SEARCH[:SETup]:ADDRESS:ADDRESS:BIT10ADDRESS?

Function Queries all I2C bus signal search 10-bit address settings.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETup]:
ADDRESS:ADDRESS:BIT10ADDRESS?
<x> = 1 to 4

:SERIALbus<x>:I2C:SEARCH[:SETup]:ADDRESS:ADDRESS:BIT10ADDRESS:DIREction

Function Sets or queries the 10-bit address data direction for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETup]:
ADDRESS:ADDRESS:BIT10ADDRESS:
DIREction {READ|WRITE|X}
:SERIALbus<x>:I2C:SEARCH[:SETup]:
ADDRESS:ADDRESS:BIT10ADDRESS:
DIREction?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT10ADDRESS:
DIRECTION READ
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT10ADDRESS:
DIRECTION?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT10ADDRESS:
DIRECTION READ

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:BIT10ADDRESS:HEXA

Function Sets the 10-bit address for I2C bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT10ADDRESS:
HEXA {<String>}
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT10ADDRESS:
HEXA "1AB"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:BIT10ADDRESS:PATTERN

Function Sets or queries the 10-bit address for I2C bus signal searching in binary notation.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT10ADDRESS:
PATTERN {<String>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT10ADDRESS:
PATTERN?

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT10ADDRESS:
PATTERN "11010111001"
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT10ADDRESS:
PATTERN?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT10ADDRESS:
PATTERN "11010111001"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS[:BIT7ADDRESS]?

Function Queries all I2C bus signal search 7-bit address settings.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS[:BIT7ADDRESS]?
<x> = 1 to 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS[:BIT7ADDRESS]:DIRECTION

Function Sets or queries the 7-bit address data direction for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS[:BIT7ADDRESS]:
DIRECTION {READ|WRITE|X}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS[:BIT7ADDRESS]:
DIRECTION?

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7ADDRESS:
DIRECTION READ
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7ADDRESS:
DIRECTION?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7ADDRESS:
DIRECTION READ

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS[:BIT7ADDRESS]:HEXA

Function Sets the 7-bit address for I2C bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS[:BIT7ADDRESS]:
HEXA {<String>}
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7ADDRESS:
HEXA "5C"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS[:BIT7ADDRESS]:PATTERN

Function Sets or queries the 7-bit address for I2C bus signal searching in binary notation.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS[:BIT7ADDRESS]:
PATTERN {<String>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS[:BIT7ADDRESS]:
PATTERN?

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7ADDRESS:
PATTERN "11100110"
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7ADDRESS:PATTERN?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7ADDRESS:
PATTERN "11100110"

5.24 SERIALbus group

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:BIT7APsub?

Function Queries all I2C bus signal search 7-bit + sub address settings.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub?
<x> = 1 to 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:BIT7APsub:ADDRESS?

Function Queries all settings related to the 7-bit address of the 7-bit + sub address for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS?
<x> = 1 to 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:BIT7APsub:ADDRESS:DIRECTION

Function Sets or queries the 7-bit + sub address data direction for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
DIRECTION {READ|WRITE|X}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
DIRECTION?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:ADDRESS:
DIRECTION READ
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:ADDRESS:
DIRECTION?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:ADDRESS:
DIRECTION READ

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:BIT7APsub:ADDRESS:HEXA

Function Sets the 7-bit address of 7-bit + sub address for I2C bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
HEXA {<String>}
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:ADDRESS:
HEXA "AB"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:BIT7APsub:ADDRESS:PATTERN

Function Sets or queries the 7-bit address of 7-bit + sub address for I2C bus signal searching in binary notation.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
PATTERN {<String>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
PATTERN?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:ADDRESS:
PATTERN "00111010"
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:ADDRESS:
PATTERN?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:ADDRESS:
PATTERN "00111010"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:BIT7APsub:SADDRESS?

Function Queries all settings related to the sub address of the 7-bit + sub address for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS?
<x> = 1 to 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:BIT7APsub:SADDRESS:HEXA

Function Sets the sub address of 7-bit + sub address for I2C bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS:
HEXA {<String>}
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:SADDRESS:
HEXA "EF"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:BIT7APsub:SADDRESS:PATTERN

Function Sets or queries the sub address of 7-bit + sub address for I2C bus signal searching in binary notation.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS:
PATTERN {<String>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS:
PATTERN?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:SADDRESS:
PATTERN "00111010"
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:SADDRESS:
PATTERN?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:BIT7APSUB:SADDRESS:
PATTERN "00111010"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:MODE

Function Sets or queries the address condition enable/disable state for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:MODE {<Boolean>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:MODE?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:MODE ON
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:MODE?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:MODE 1

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:ADDRESS:TYPE

Function Sets or queries the address condition address format for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:TYPE {BIT10ADDRESS|
BIT7ADDRESS|BIT7APsub}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:ADDRESS:TYPE?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:TYPE BIT10ADDRESS
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:TYPE?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:ADDRESS:TYPE BIT10ADDRESS

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:DATA?

Function Queries all settings related to the data of the I2C bus signal search address pattern.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:DATA?
<x> = 1 to 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:DATA:BCOUNT

Function Sets or queries the data pattern comparison position for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:DATA:BCOUNT {<NRF>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:DATA:BCOUNT?
<x> = 1 to 4
<NRF> = 0 to 9999

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:BCOUNT 0
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:BCOUNT?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:BCOUNT 0

:SERIALbus<x>:I2C:SEARCH[:SETUP]:ADDRESS:DATA:BMODE

Function Sets or queries on/off state of the data pattern comparison position for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:DATA:BMODE {<Boolean>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
ADDRESS:DATA:BMODE?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:BMODE ON
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:BMODE?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:BMODE 1

5.24 SERIALbus group

:SERIALbus<x>:I2C:SEARCh[:SETup]:ADDRESS:DATA:CONDition

Function Sets or queries the data judgment method (true/false) for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCh[:SETup]:
ADDRESS:DATA:CONDition {FALSE|TRUE}
:SERIALbus<x>:I2C:SEARCh[:SETup]:
ADDRESS:DATA:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:CONDITION FALSE
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:CONDITION?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:CONDITION FALSE

:SERIALbus<x>:I2C:SEARCh[:SETup]:ADDRESS:DATA:DBYTe

Function Sets or queries the number of reference values for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCh[:SETup]:
ADDRESS:DATA:DBYTe {<Nrf>}
:SERIALbus<x>:I2C:SEARCh[:SETup]:
ADDRESS:DATA:DBYTe?
<x> = 1 to 4
<Nrf> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:DBYTE 1
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:DBYTE?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:DBYTE 1

:SERIALbus<x>:I2C:SEARCh[:SETup]:ADDRESS:DATA:DPATtern?

Function Queries all data settings for the data conditions for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCh[:SETup]:
ADDRESS:DATA:DPATtern?
<x> = 1 to 4

:SERIALbus<x>:I2C:SEARCh[:SETup]:ADDRESS:DATA:DPATtern:HEXa

Function Sets the data condition data for I2C bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:I2C:SEARCh[:SETup]:
ADDRESS:DATA:DPATtern:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:DPATTERN:HEXA "AB"

Description The number of bytes that can be set varies depending on the :SERIALbus<x>:I2C:SEARCh[:SETup]:ADDRESS:DATA:DBYTe setting.

:SERIALbus<x>:I2C:SEARCh[:SETup]:ADDRESS:DATA:DPATtern:PATtern

Function Sets or queries the data condition data for I2C bus signal searching in binary notation.

Syntax :SERIALbus<x>:I2C:SEARCh[:SETup]:
ADDRESS:DATA:DPATtern:
PATtern {<String>}
:SERIALbus<x>:I2C:SEARCh[:SETup]:
ADDRESS:DATA:DPATtern:PATtern?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:DPATTERN:
PATTERN "10001101"
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:DPATTERN:PATTERN?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:DPATTERN:
PATTERN "10001101"

Description The number of bytes that can be set varies depending on the :SERIALbus<x>:I2C:SEARCh[:SETup]:ADDRESS:DATA:DBYTe setting.

:SERIALbus<x>:I2C:SEARCh[:SETup]:ADDRESS:DATA:MODE

Function Sets or queries the data condition enable/disable state for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCh[:SETup]:
ADDRESS:DATA:MODE {<Boolean>}
:SERIALbus<x>:I2C:SEARCh[:SETup]:
ADDRESS:DATA:MODE?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:MODE ON
:SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:MODE?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
ADDRESS:DATA:MODE 1

:SERIALbus<x>:I2C:SEARCh[:SETup]:GENERALcall?

Function Queries all I2C bus signal search general call settings.

Syntax :SERIALbus<x>:I2C:SEARCh[:SETup]:
GENERALcall?
<x> = 1 to 4

:SERIALbus<x>:I2C:SEARCh[:SETup]:GENERALcall:BIT7Maddress?

Function Queries all settings related to the 7-bit master address of general call for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCh[:SETup]:
GENERALcall:BIT7Maddress?
<x> = 1 to 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:GENERALCALL:BIT7MAddress:HEXA

Function Sets the 7-bit master address of general call for I2C bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
GENERALCALL:BIT7MAddress:HEXA{<String>}
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
GENERALCALL:BIT7MADDRESS:HEXA "AB"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:GENERALCALL:BIT7MAddress:PATTERN

Function Sets or queries the 7-bit master address of general call for I2C bus signal searching in binary notation.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
GENERALCALL:BIT7MAddress:
PATTERN{<String>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
GENERALCALL:BIT7MAddress:PATTERN?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
GENERALCALL:BIT7MADDRESS:
PATTERN "0010110"
:SERIALBUS1:I2C:SEARCH:SETUP:
GENERALCALL:BIT7MADDRESS:PATTERN?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
GENERALCALL:BIT7MADDRESS:
PATTERN "0010110"

:SERIALbus<x>:I2C:SEARCH[:SETUP]:GENERALCALL:SBYTE

Function Sets or queries the general call second byte type for I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
GENERALCALL:SBYTE {BIT7MAddress|
DONTcare|H04|H06}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
GENERALCALL:SBYTE?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
GENERALCALL:SBYTE BIT7MADDRESS
:SERIALBUS1:I2C:SEARCH:SETUP:
GENERALCALL:SBYTE?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
GENERALCALL:SBYTE BIT7MADDRESS

:SERIALbus<x>:I2C:SEARCH[:SETUP]:MODE

Function Sets or queries the I2C bus signal search mode setting.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
MODE {ADRData|EVERystart|
GENERALCALL|HSMODE|NONAck|STARTbyte}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
MODE?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:
MODE ADRDATA
:SERIALBUS1:I2C:SEARCH:SETUP:MODE?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
MODE ADRDATA

:SERIALbus<x>:I2C:SEARCH[:SETUP]:NONAck?

Function Queries all I2C bus signal search NON ACK ignore mode settings.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
NONAck?
<x> = 1 to 4

:SERIALbus<x>:I2C:SEARCH[:SETUP]:NONAck:HSMODE

Function Sets or queries whether NONACK is to be ignored in high-speed mode of I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCH[:SETUP]:
NONAck:HSMODE {<Boolean>}
:SERIALbus<x>:I2C:SEARCH[:SETUP]:
NONAck:HSMODE?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:NONACK:
HSMODE ON
:SERIALBUS1:I2C:SEARCH:SETUP:NONACK:
HSMODE?
-> :SERIALBUS1:I2C:SEARCH:SETUP:
NONACK:HSMODE 1

5.24 SERIALbus group

:SERIALbus<x>:I2C:SEARCh[:SETup]:NONAck:READAccess

Function Sets or queries whether NONACK is to be ignored in read access mode of I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCh[:SETup]:NONAck:READAccess {<Boolean>}
:SERIALbus<x>:I2C:SEARCh[:SETup]:NONAck:READAccess?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:NONACK:READACCESS ON
:SERIALBUS1:I2C:SEARCH:SETUP:NONACK:READACCESS?
-> :SERIALBUS1:I2C:SEARCH:SETUP:NONACK:READACCESS 1

:SERIALbus<x>:I2C:SEARCh[:SETup]:NONAck:STARTbyte

Function Sets or queries whether NONACK is to be ignored in the start byte of I2C bus signal searching.

Syntax :SERIALbus<x>:I2C:SEARCh[:SETup]:NONAck:STARTbyte {<Boolean>}
:SERIALbus<x>:I2C:SEARCh[:SETup]:NONAck:STARTbyte?
<x> = 1 to 4

Example :SERIALBUS1:I2C:SEARCH:SETUP:NONACK:STARTBYTE ON
:SERIALBUS1:I2C:SEARCH:SETUP:NONACK:STARTBYTE?
-> :SERIALBUS1:I2C:SEARCH:SETUP:NONACK:STARTBYTE 1

:SERIALbus<x>:LIN?

Function Queries all LIN bus signal analysis and search settings.

Syntax :SERIALbus<x>:LIN?
<x> = 1 to 4

:SERIALbus<x>:LIN:ANALyze?

Function Queries all LIN bus signal analysis settings.

Syntax :SERIALbus<x>:LIN:ANALyze?
<x> = 1 to 4

:SERIALbus<x>:LIN[:ANALyze]:SETup?

Function Queries all LIN bus signal analysis bus settings.

Syntax :SERIALbus<x>:LIN[:ANALyze]:SETup?
<x> = 1 to 4

:SERIALbus<x>:LIN[:ANALyze]:SETup:BRATe

Function Sets or queries the LIN bus signal analysis bit rate (data transfer rate).

Syntax :SERIALbus<x>:LIN[:ANALyze]:SETup:BRATe {<Nrf>|USER,<Nrf>}
:SERIALbus<x>:LIN[:ANALyze]:SETup:BRATe?
<x> = 1 to 4

<Nrf> = 1200, 2400, 4800, 9600, 19200
USER <Nrf> = See the Features Guide for this information.

Example :SERIALBUS1:LIN:ANALYZE:SETUP:BRATE 2400
:SERIALBUS1:LIN:ANALYZE:SETUP:BRATE?
-> :SERIALBUS1:LIN:ANALYZE:SETUP:BRATE 2400

:SERIALbus<x>:LIN[:ANALyze]:SETup:REViSiOn

Function Sets or queries the LIN bus signal analysis revision number.

Syntax :SERIALbus<x>:LIN[:ANALyze]:SETup:REViSiOn {BOTH|LIN1 _ 3|LIN2 _ 0}
:SERIALbus<x>:LIN[:ANALyze]:SETup:REViSiOn?
<x> = 1 to 4

Example :SERIALBUS1:LIN:ANALYZE:SETUP:REVISION LIN1 _ 3
:SERIALBUS1:LIN:ANALYZE:SETUP:REVISION?
-> :SERIALBUS1:LIN:ANALYZE:SETUP:REVISION LIN1 _ 3

:SERIALbus<x>:LIN[:ANALyze]:SETup:SOURce

Function Sets or queries the LIN bus signal analysis source.

Syntax :SERIALbus<x>:LIN[:ANALyze]:SETup:SOURce {<Nrf>|MATH<y>}
:SERIALbus<x>:LIN[:ANALyze]:SETup:SOURce?
<x> = 1 to 4
<Nrf> = 1 to 4
MATH<y>'s <y> = 1 to 4

Example :SERIALBUS1:LIN:ANALYZE:SETUP:SOURCE 1
:SERIALBUS1:LIN:ANALYZE:SETUP:SOURCE?
-> :SERIALBUS1:LIN:ANALYZE:SETUP:SOURCE 1

**:SERIALbus<x>:LIN[:ANALyze]:SETup:SPOi
nt**

Function Sets or queries the LIN bus signal analysis sample point.

Syntax :SERIALbus<x>:LIN[:ANALyze]:SETup:
SPOint {<NRf>}
:SERIALbus<x>:LIN[:ANALyze]:SETup:
SPOint?
<x> = 1 to 4
<NRf> = 18.8 to 90.6

Example :SERIALBUS1:LIN:ANALYZE:SETUP:
SPOINT 18.8
:SERIALBUS1:LIN:ANALYZE:SETUP:
SPOINT?
-> :SERIALBUS1:LIN:ANALYZE:SETUP:
SPOINT 18.8

:SERIALbus<x>:LIN:DETail?

Function Queries all LIN bus signal analysis result list settings.

Syntax :SERIALbus<x>:LIN:DETail?
<x> = 1 to 4

:SERIALbus<x>:LIN:DETail:DISPlay

Function Sets or queries the display mode for the LIN bus signal analysis result list.

Syntax :SERIALbus<x>:LIN:DETail:
DISPlay {FULl|LOWer|UPPer}
:SERIALbus<x>:LIN:DETail:DISPlay?
<x> = 1 to 4

Example :SERIALBUS1:LIN:DETAIL:DISPLAY FULL
:SERIALBUS1:LIN:DETAIL:DISPLAY?
-> :SERIALBUS1:LIN:DETAIL:
DISPLAY FULL

:SERIALbus<x>:LIN:DETail:LIST:ITEM?

Function Queries all items that will be displayed in the LIN bus signal analysis result list.

Syntax :SERIALbus<x>:LIN:DETail:LIST:ITEM?
<x> = 1 to 4

Example :SERIALBUS1:LIN:DETAIL:LIST:ITEM?
-> :SERIALBUS1:LIN:DETAIL:LIST:
ITEM "No.,Time(ms),ID,ID-Field,Data,
Checksum,Information"

:SERIALbus<x>:LIN:DETail:LIST:VALue?

Function Queries all the data for the specified analysis number in the LIN bus signal analysis result list.

Syntax :SERIALbus<x>:LIN:DETail:LIST:
VALue? {<NRf>|MAXimum|MINimum}
<x> = 1 to 4
<NRf> = See the Features Guide for this information.

Example :SERIALBUS1:LIN:DETAIL:LIST:VALUE? 0
-> :SERIALBUS1:LIN:DETAIL:LIST:
VALUE "0,-1,0016,30,F0,2D04,CE,,,"

:SERIALbus<x>:LIN:SEARCh?

Function Queries all LIN bus signal search settings.

Syntax :SERIALbus<x>:LIN:SEARCh?
<x> = 1 to 4

:SERIALbus<x>:LIN:SEARCh:ABORT

Function Aborts the LIN bus signal search.

Syntax :SERIALbus<x>:LIN:SEARCh:ABORT
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:ABORT

:SERIALbus<x>:LIN:SEARCh:EXECute

Function Executes a LIN bus signal search.

Syntax :SERIALbus<x>:LIN:SEARCh:EXECute
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:EXECUTE

:SERIALbus<x>:LIN:SEARCh:FJUMp:BREAK

Function Jumps to the break field in the LIN bus signal search result.

Syntax :SERIALbus<x>:LIN:SEARCh:FJUMp:BREAK
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:FJUMP:BREAK

:SERIALbus<x>:LIN:SEARCh:FJUMp:CSUM

Function Jumps to the checksum field in the LIN bus signal search result.

Syntax :SERIALbus<x>:LIN:SEARCh:FJUMp:CSUM
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:FJUMP:CSUM

:SERIALbus<x>:LIN:SEARCh:FJUMp:DATA

Function Jumps to the data field in the LIN bus signal search result.

Syntax :SERIALbus<x>:LIN:SEARCh:FJUMp:DATA
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:FJUMP:DATA

5.24 SERIALbus group

:SERIALbus<x>:LIN:SEARCH:FJUMP:IDENTifier

Function Jumps to the identifier field in the LIN bus signal search result.

Syntax :SERIALbus<x>:LIN:SEARCH:FJUMP:
IDENTifier
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:FJUMP:
IDENTIFIER

:SERIALbus<x>:LIN:SEARCH:FJUMP:SYNCh

Function Jumps to the sync field in the LIN bus signal search result.

Syntax :SERIALbus<x>:LIN:SEARCH:FJUMP:SYNCh
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:FJUMP:SYNCH

:SERIALbus<x>:LIN:SEARCH:SELEct

Function Sets which detected point to display in the LIN bus signal search zoom window and queries the zoom position of the detected point.

Syntax :SERIALbus<x>:LIN:SEARCH:
SElect {<NRF>|MAXimum}
:SERIALbus<x>:LIN:SEARCH:SElect?
<x> = 1 to 4
<NRF> = 0 to 49999

Example :SERIALBUS1:LIN:SEARCH:SELECT 1
:SERIALBUS1:LIN:SEARCH:SELECT?
-> :SERIALBUS1:LIN:SEARCH:
SELECT 1.50000000

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:LIN:SEARCH:SELEct? MAXimum

Function Queries the number of detected points in the LIN bus signal search.

Syntax :SERIALbus<x>:LIN:SEARCH:
SElect? {MAXimum}
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:
SELECT? MAXIMUM
-> :SERIALBUS1:LIN:SEARCH:SELECT 100

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:LIN:SEARCH:SETup?

Function Queries all LIN bus signal search condition settings.

Syntax :SERIALbus<x>:LIN:SEARCH:SETup?
<x> = 1 to 4

:SERIALbus<x>:LIN:SEARCH:SETup:ERROR?

Function Queries all LIN bus signal search error settings.

Syntax :SERIALbus<x>:LIN:SEARCH:SETup:
ERROR?
<x> = 1 to 4

:SERIALbus<x>:LIN:SEARCH:SETup:ERROR:CHECKsum

Function Sets or queries the LIN bus signal search checksum error setting.

Syntax :SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:CHECKsum {<Boolean>}
:SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:CHECKsum?
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
CHECKSUM ON
:SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
CHECKSUM?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
ERROR:CHECKSUM 1

:SERIALbus<x>:LIN:SEARCH:SETup:ERROR:FRAMing

Function Sets or queries the LIN bus signal search framing error setting.

Syntax :SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:FRAMing {<Boolean>}
:SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:FRAMing?
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
FRAMING ON
:SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
FRAMING?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
ERROR:FRAMING 1

:SERIALbus<x>:LIN:SEARCH:SETup:ERROR:PARity

Function Sets or queries the LIN bus signal search parity error setting.

Syntax :SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:PARity {<Boolean>}
:SERIALbus<x>:LIN:SEARCH:SETup:
ERROR:PARity?
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
PARITY ON
:SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
PARITY?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
ERROR:PARITY 1

:SERIALbus<x>:LIN:SEARCh:SETup:ERRor:SYNCh

Function Sets or queries the LIN bus signal search synch error setting.

Syntax :SERIALbus<x>:LIN:SEARCh:SETup:
ERRor:SYNCh {<Boolean>}
:SERIALbus<x>:LIN:SEARCh:SETup:
ERRor:SYNCh?

Example :SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
SYNCH ON
:SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
SYNCH?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
ERROR:SYNCH 1

:SERIALbus<x>:LIN:SEARCh:SETup:ERRor:TIMEout

Function Sets or queries the LIN bus signal search timeout error setting.

Syntax :SERIALbus<x>:LIN:SEARCh:SETup:
ERRor:TIMEout {<Boolean>}
:SERIALbus<x>:LIN:SEARCh:SETup:
ERRor:TIMEout?

Example :SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
TIMEOUT ON
:SERIALBUS1:LIN:SEARCH:SETUP:ERROR:
TIMEOUT?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
ERROR:TIMEOUT 1

:SERIALbus<x>:LIN:SEARCh:SETup:IDData?

Function Queries all ID and data condition settings for LIN bus signal searching.

Syntax :SERIALbus<x>:LIN:SEARCh:SETup:
IDData?
<x> = 1 to 4

:SERIALbus<x>:LIN:SEARCh:SETup:IDData:DATA?

Function Queries all LIN bus signal search data settings.

Syntax :SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA?
<x> = 1 to 4

:SERIALbus<x>:LIN:SEARCh:SETup:IDData:DATA:CONDition

Function Sets or queries the LIN bus signal search data condition setting.

Syntax :SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA:CONDition {EQUAL|FALSE|
GREATER|INSIDE|LESS|NOTEQUAL|
OUTSIDE|TRUE}
:SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA:CONDition?

Example :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:CONDITION EQUAL
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:CONDITION?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:CONDITION EQUAL

:SERIALbus<x>:LIN:SEARCh:SETup:IDData:DATA:DBYTE

Function Sets or queries the number of data bytes for LIN bus signal searching.

Syntax :SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA:DBYTE {<NRF>}
:SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA:DBYTE?

Example :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:DBYTE 1
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:DBYTE?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:DBYTE 1

:SERIALbus<x>:LIN:SEARCh:SETup:IDData:DATA:DECimal<y>

Function Sets or queries the data for LIN bus signal searching in decimal notation.

Syntax :SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA:DECimal<y> {<NRF>}
:SERIALbus<x>:LIN:SEARCh:SETup:
IDData:DATA:DECimal<y>?

Example :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:DECIMAL1 1
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:DECIMAL1?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:DECIMAL1 1

5.24 SERIALbus group

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:DATA:DPATtern?

Function Queries all LIN bus signal search data settings.

Syntax :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:DPATtern?
<x> = 1 to 4

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:DATA:DPATtern:HEXa

Function Sets the data for LIN bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:DPATtern:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:DPATTERN:HEXA "12"

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:DATA:DPATtern:PATtern

Function Sets or queries the data for LIN bus signal searching in binary notation.

Syntax :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:DPATtern:
PATtern {<String>}
:SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:DPATtern:PATtern?
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:DPATTERN:PATTERN "00110101"
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:DPATTERN:PATTERN?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:DPATTERN:
PATTERN "00110101"

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:DATA:ENDian

Function Sets or queries the data endian for LIN bus signal searching.

Syntax :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:ENDian {BIG|LITtle}
:SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:ENDian?
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:ENDIAN BIG
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:ENDIAN?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:ENDIAN BIG

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:DATA:MODE

Function Sets or queries the data condition enable/disable state for LIN bus signal searching.

Syntax :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:MODE {<Boolean>}
:SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:MODE?
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:MODE ON
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:MODE?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:MODE 1

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:DATA:MSBLsb

Function Sets or queries the MSB/LSB bit for LIN bus signal searching.

Syntax :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:MSBLsb {<Nrf>,<Nrf>}
:SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:MSBLsb?
<x> = 1 to 4
<Nrf> = 0 to 63

Example :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:MSBLSB 1,0
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:MSBLSB?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:MSBLSB 1,0

:SERIALbus<x>:LIN:SEARCH:SETup:IDData:DATA:SIGN

Function Sets or queries the data sign for LIN bus signal searching.

Syntax :SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:SIGN {SIGN|UNSign}
:SERIALbus<x>:LIN:SEARCH:SETup:
IDData:DATA:SIGN?
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:SIGN SIGN
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
DATA:SIGN?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:DATA:SIGN SIGN

:SERialbus<x>:LIN:SEARCh:SETup:IDData:IDENTifier?

Function Queries all LIN bus signal search identifier settings.

Syntax :SERialbus<x>:LIN:SEARCh:SETup:
IDData:IDENTifier?
<x> = 1 to 4

:SERialbus<x>:LIN:SEARCh:SETup:IDData:IDENTifier:ID?

Function Queries all LIN bus signal search ID settings.

Syntax :SERialbus<x>:LIN:SEARCh:SETup:
IDData:IDENTifier:ID?
<x> = 1 to 4

:SERialbus<x>:LIN:SEARCh:SETup:IDData:IDENTifier:ID:HEXa

Function Sets the ID for LIN bus signal searching in hexadecimal notation.

Syntax :SERialbus<x>:LIN:SEARCh:SETup:
IDData:IDENTifier:ID:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
IDENTIFIER:ID:HEXa "1E"

:SERialbus<x>:LIN:SEARCh:SETup:IDData:IDENTifier:ID:MODE

Function Sets or queries the ID condition enable/disable state for LIN bus signal searching.

Syntax :SERialbus<x>:LIN:SEARCh:SETup:
IDData:IDENTifier:ID:
MODE {<Boolean>}
:SERialbus<x>:LIN:SEARCh:SETup:
IDData:IDENTifier:ID:MODE?
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
IDENTIFIER:ID:MODE ON
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
IDENTIFIER:ID:MODE?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:IDENTIFIER:ID:MODE 1

:SERialbus<x>:LIN:SEARCh:SETup:IDData:IDENTifier:ID:PATtern

Function Sets or queries the ID for LIN bus signal searching in binary notation.

Syntax :SERialbus<x>:LIN:SEARCh:SETup:
IDData:IDENTifier:ID:PATtern{<String>}
:SERialbus<x>:LIN:SEARCh:SETup:
IDData:IDENTifier:ID:PATtern?
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
IDENTIFIER:ID:PATTERN "101100"
:SERIALBUS1:LIN:SEARCH:SETUP:IDDATA:
IDENTIFIER:ID:PATTERN?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
IDDATA:IDENTIFIER:ID:
PATTERN "101100"

:SERialbus<x>:LIN:SEARCh:SETup:MODE

Function Sets or queries the LIN bus signal search mode setting.

Syntax :SERialbus<x>:LIN:SEARCh:SETup:
MODE {BSYNCh|ERRor|IDData}
:SERialbus<x>:LIN:SEARCh:SETup:MODE?
<x> = 1 to 4

Example :SERIALBUS1:LIN:SEARCH:SETUP:
MODE BSYNCH
:SERIALBUS1:LIN:SEARCH:SETUP:MODE?
-> :SERIALBUS1:LIN:SEARCH:SETUP:
MODE BSYNCH

:SERialbus<x>:PSI5?

Function Queries all PSI5 signal analysis and search settings.

Syntax :SERialbus<x>:PSI5?
<x> = 1 to 4

:SERialbus<x>:PSI5:ANALyze?

Function Queries all PSI5 signal analysis settings.

Syntax :SERialbus<x>:PSI5:ANALyze?
<x> = 1 to 4

:SERialbus<x>:PSI5[:ANALyze]:SETup?

Function Queries all bus setup settings for PSI5 signal analysis.

Syntax :SERialbus<x>:PSI5[:ANALyze]:SETup?
<x> = 1 to 4

5.24 SERIALbus group

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA?

Function Queries all PSI5 signal analysis data signal settings.

Syntax :SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA?
<x> = 1 to 4

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:BRATe

Function Sets or queries the PSI5 signal analysis bit rate (data transfer rate).

Syntax :SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:BRATe {<NRf>|USER,<NRf>}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:BRATe?
<x> = 1 to 4

<NRf> = 125000, 189000
USER <NRf> = 10000 to 1000000

Example :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:BRATE 125000
:SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:BRATE?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:BRATE 125000

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:CTOLerance

Function Sets or queries the PSI5 signal analysis clock tolerance range.

Syntax :SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:CTOLerance {<NRf>}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:CTOLerance?
<x> = 1 to 4

<NRf> = $\pm 0.5\%$ to $\pm 33.3\%$, resolution: 0.1%

Example :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:CTOLERANCE 5
:SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:CTOLERANCE?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:CTOLERANCE 5.00E+00

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:DBITs

Function Sets or queries the data length for PSI5 signal analysis.

Syntax :SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:DBITs {<NRf>}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:DBITs?
<x> = 1 to 4
<NRf> = 10, 16

Example :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:DBITS 10
:SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:DBITS?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:DBITS 10

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:EDETection

Function Sets or queries the error detection method for PSI5 signal analysis.

Syntax :SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:EDETection {CRC|PARity}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:EDETection?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:EDETECTION CRC
:SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:EDETECTION?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:EDETECTION CRC

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SNRejection?

Function Queries all PSI5 signal analysis sync noise rejection settings.

Syntax :SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SNRejection?
<x> = 1 to 4

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SNRejection:ETIME

Function Sets or queries the rejection end time for PSI5 signal analysis sync noise rejection.

Syntax :SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SNRejection:ETIME {<NRf>}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SNRejection:ETIME?
<x> = 1 to 4
<NRf> = 0 to 0.02 s, 0.1 us steps

Example :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:SNREJECTION:ETIME 44.0000E-06
:SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:SNREJECTION:ETIME?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:SNREJECTION:ETIME 44.0000E-06

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SNRejection:MODE

Function Sets or queries the on/off state of PSI5 signal analysis sync noise rejection.

Syntax :SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SNRejection:MODE {<Boolean>}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SNRejection:MODE?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:SNREJECTION:MODE ON
:SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:SNREJECTION:MODE?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:SNREJECTION:MODE 1

:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SOURce

Function Sets or queries the data source for PSI5 signal analysis.

Syntax :SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SOURce {<NRf>|MATH<y>}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:DATA:SOURce?
<x> = 1 to 4
<NRf> = 1 to 4
MATH<y>'s <y> = 1 to 4

Example :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:SOURCE 1
:SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:SOURCE?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:DATA:SOURCE 1

:SERIALbus<x>:PSI5[:ANALyze]:SETup:NUMBERofslot

Function Sets or queries the number of slots for PSI5 signal analysis.

Syntax :SERIALbus<x>:PSI5[:ANALyze]:SETup:NUMBERofslot {AUTO|<NRf>}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:NUMBERofslot?
<x> = 1 to 4
<NRf> = 1 to 6

Example :SERIALBUS1:PSI5:ANALYZE:SETUP:NUMBEROFSLOT AUTO
:SERIALBUS1:PSI5:ANALYZE:SETUP:NUMBEROFSLOT?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:NUMBEROFSLOT AUTO

:SERIALbus<x>:PSI5[:ANALyze]:SETup:SLOT<y>

Function Sets or queries the slot's start and end times for PSI5 signal analysis.

Syntax :SERIALbus<x>:PSI5[:ANALyze]:SETup:SLOT<y> {<NRf>[,<NRf>]}
:SERIALbus<x>:PSI5[:ANALyze]:SETup:SLOT<y>?
<x> = 1 to 4
<y> = 1 to 6 (slot number)

Example :SERIALBUS1:PSI5:ANALYZE:SETUP:SLOT6 870.000E-06,1.08830E-03
:SERIALBUS1:PSI5:ANALYZE:SETUP:SLOT6?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:SLOT6 870.000E-06,1.08830E-03

Description The first <NRf> is the slot's start time and the second the end time.

If :SERIALbus<x>:PSI5[:ANALyze]:SETup:NUMBERofslot is a number from 1 to 6, when the last slot number is specified with <y>, the second <NRf> becomes valid. If :SERIALbus<x>:PSI5[:ANALyze]:SETup:NUMBERofslot is AUTO, query is not possible.

5.24 SERIALbus group

:SERIALbus<x>:PSI5[:ANALyze]:SETup:SYNC

Function Sets or queries the sync signal for PSI5 signal analysis.

Syntax :SERIALbus<x>:PSI5[:ANALyze]:SETup:
SYNC {<NRf>|MATH<y>|NONE}
:SERIALbus<x>:PSI5:ANALyze:SETup:
SYNC?
<x> = 1 to 4
<NRf> = 1 to 4
MATH<y>'s <y> = 1 to 4

Example :SERIALBUS1:PSI5:ANALYZE:SETUP:
SYNC 1
:SERIALBUS1:PSI5:ANALYZE:SETUP:SYNC?
-> :SERIALBUS1:PSI5:ANALYZE:SETUP:
SYNC 1

:SERIALbus<x>:PSI5:DETail?

Function Queries all PSI5 signal analysis list settings.

Syntax :SERIALbus<x>:PSI5:DETail?
<x> = 1 to 4

:SERIALbus<x>:PSI5:DETail:DISPlay

Function Sets or queries the display mode for the PSI5 signal analysis list.

Syntax :SERIALbus<x>:PSI5:DETail:
DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:PSI5:DETail:DISPlay?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:DETAIL:DISPLAY FULL
:SERIALBUS1:PSI5:DETAIL:DISPLAY?
-> :SERIALBUS1:PSI5:DETAIL:
DISPLAY FULL

:SERIALbus<x>:PSI5:DETail:LIST:ALL?

Function Queries all the data for all analysis numbers in the PSI5 signal analysis result list.

Syntax :SERIALbus<x>:PSI5:DETail:LIST:ALL?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:DETAIL:LIST:ALL?
-> #8 (8-digit number of bytes)(data sequence)

Description All the data for all analysis numbers are returned in <block data> format. The data of each analysis number is separated by ASCII code 0AH.

:SERIALbus<x>:PSI5:DETail:LIST:ITEM?

Function Queries all items that will be displayed in the PSI5 signal analysis result list.

Syntax :SERIALbus<x>:PSI5:DETail:LIST:ITEM?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:DETAIL:LIST:ITEM?
-> :SERIALBUS1:PSI5:DETAIL:LIST:
ITEM "No.Time(ms)Slot No.DataParity/
CRCInformation"

:SERIALbus<x>:PSI5:DETail:LIST:VALue?

Function Queries all the data for the specified analysis number in the PSI5 signal analysis result list.

Syntax :SERIALbus<x>:PSI5:DETail:LIST:
VALue? {<NRf>|MAXimum|MINimum}
<x> = 1 to 4
<NRf> = -400000 to 400000

Example :SERIALBUS1:PSI5:DETAIL:LIST:
VALUE? 1
-> :SERIALBUS1:PSI5:DETAIL:LIST:
VALUE "1,0.062104,1,AB67,2,,,"

:SERIALbus<x>:PSI5:SEARCh?

Function Queries all PSI5 signal search settings.

Syntax :SERIALbus<x>:PSI5:SEARCh?
<x> = 1 to 4

:SERIALbus<x>:PSI5:SEARCh:ABORT

Function Aborts the PSI5 signal search.

Syntax :SERIALbus<x>:PSI5:SEARCh:ABORT
<x> = 1 to 4

Example :SERIALBUS1:PSI5:SEARCH:ABORT

:SERIALbus<x>:PSI5:SEARCh:EXECute

Function Executes a PSI5 signal search.

Syntax :SERIALbus<x>:PSI5:SEARCh:EXECute
<x> = 1 to 4

Example :SERIALBUS1:PSI5:SEARCH:EXECUTE

:SERIALbus<x>:PSI5:SEARCH:SELEct

Function Sets which detected point to display in the PSI5 signal search zoom window and queries the zoom position of the detected point.

Syntax :SERIALbus<x>:PSI5:SEARCH:
SELEct {<NRf>|MAXimum}
:SERIALbus<x>:PSI5:SEARCH:SELEct?
<x> = 1 to 4
<NRf> = 0 to 49999

Example :SERIALBUS1:PSI5:SEARCH:SELECT 1
:SERIALBUS1:PSI5:SEARCH:SELECT?
-> :SERIALBUS1:PSI5:SEARCH:
SELECT 1.50000000

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:PSI5:SEARCH:SETup?

Function Queries all PSI5 signal search setup settings.

Syntax :SERIALbus<x>:PSI5:SEARCH:SETup?
<x> = 1 to 4

:SERIALbus<x>:PSI5:SEARCH[:SETup]:DATA?

Function Queries all PSI5 signal data search settings.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETup]:
DATA?
<x> = 1 to 4

:SERIALbus<x>:PSI5:SEARCH[:SETup]:DATA:DATA?

Function Queries all data settings of the PSI5 signal data search.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETup]:
DATA:DATA?
<x> = 1 to 4

:SERIALbus<x>:PSI5:SEARCH[:SETup]:DATA:DATA:CONDition

Function Sets or queries the comparison condition of the PSI5 signal data search.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETup]:
DATA:DATA:CONDition {EQUAL|FALSE|
GREATER|INSide|LESS|NOTEQUAL|
OUTSide|TRUE}
:SERIALbus<x>:PSI5:SEARCH[:SETup]:
DATA:DATA:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:CONDITION EQUAL
:SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:CONDITION?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
DATA:DATA:CONDITION EQUAL

:SERIALbus<x>:PSI5:SEARCH[:SETup]:DATA:DATA:DECimal<y>

Function Sets or queries the data of the PSI5 signal data search in decimal notation.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETup]:
DATA:DATA:DECimal<y> {<NRf>}
:SERIALbus<x>:PSI5:SEARCH[:SETup]:
DATA:DATA:DECimal<y>?
<x> = 1 to 4
<y> = 1 to 2

When <NRf> = 10 bit data, -512 to 511

When <NRf> = 16 bit data, -32768 to 32767

Example :SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:DECIMAL1 -10
:SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:DECIMAL1?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
DATA:DATA:DECIMAL1 -10

Description This command is valid when :SERIALbus<x>:PSI5:SEARCH[:SETup]:DATA:DATA:CONDition is EQUAL, GREATER, INSide, LESS, NOTEQUAL, or OUTSide.

:SERIALbus<x>:PSI5:SEARCH[:SETup]:DATA:DATA:HEXa

Function Sets the data of the PSI5 signal data search in hexadecimal notation.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETup]:
DATA:DATA:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:HEXa "3A0F"

Description This command is valid when :SERIALbus<x>:PSI5:SEARCH:SETup:DATA:DATA:CONDition is TRUE or FALSE.

:SERIALbus<x>:PSI5:SEARCH[:SETup]:DATA:DATA:PATtern

Function Sets or queries the data of the PSI5 signal data search in binary notation.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETup]:
DATA:DATA:PATtern {<String>}
:SERIALbus<x>:PSI5:SEARCH[:SETup]:
DATA:DATA:PATtern?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:PATTERN "0011010100001111"
:SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
DATA:PATTERN?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
DATA:DATA:PATTERN "0011010100001111"

Description This command is valid when :SERIALbus<x>:PSI5:SEARCH:SETup:DATA:DATA:CONDition is TRUE or FALSE.

5.24 SERIALbus group

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:DATA:FRAMEinslot?

Function Queries all slot specification settings of the PSI5 signal data search.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETUP]:
DATA:FRAMEinslot?
<x> = 1 to 4

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:DATA:FRAMEinslot:MODE

Function Sets or queries the slot specification condition (enabled/disabled) of the PSI5 signal data search.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETUP]:
DATA:FRAMEinslot:MODE {<Boolean>}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
DATA:FRAMEinslot:MODE?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
FRAMEINSLOT:MODE ON
:SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
FRAMEINSLOT:MODE?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
DATA:FRAMEINSLOT:MODE 1

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:DATA:FRAMEinslot:SNUMBER

Function Sets or queries the slot number of the PSI5 signal data search.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETUP]:
DATA:FRAMEinslot:SNUMBER {<Nrf>}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
DATA:FRAMEinslot:SNUMBER?
<x> = 1 to 4
<Nrf> = 1 to 6

Example :SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
FRAMEINSLOT:SNUMBER 1
:SERIALBUS1:PSI5:SEARCH:SETUP:DATA:
FRAMEINSLOT:SNUMBER?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
DATA:FRAMEINSLOT:SNUMBER 1

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:ERROR?

Function Queries all PSI5 signal error search settings.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR?
<x> = 1 to 4

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:ERROR:FNUMBER

Function Sets or queries the PSI5 signal frame number error search.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:FNUMBER {<Boolean>}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:FNUMBER?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
FNUMBER ON
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
FNUMBER?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
ERROR:FNUMBER 1

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:ERROR:FRAME

Function Sets or queries the PSI5 signal frame error search.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:FRAME {<Boolean>}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:FRAME?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
FRAME ON
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
FRAME?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
ERROR:FRAME 1

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:ERROR:MANCHESTER

Function Sets or queries the PSI5 signal Manchester error search.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:MANCHESTER {<Boolean>}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:MANCHESTER?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
MANCHESTER ON
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
MANCHESTER?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
ERROR:MANCHESTER 1

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:ERROR:PCRC

Function Sets or queries the PSI5 signal parity/CRC error search.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:PCRC {<Boolean>}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:PCRC?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
PCRC ON
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
PCRC?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
ERROR:PCRC 1

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:ERROR:SBIT

Function Sets or queries the PSI5 signal start bit error search.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:SBIT {<Boolean>}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:SBIT?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
SBIT ON
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
SBIT?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
ERROR:SBIT 1

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:ERROR:SBOundary

Function Sets or queries the PSI5 signal slot boundary error search.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:SBOundary {<Boolean>}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
ERROR:SBOundary?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
SBOundary ON
:SERIALBUS1:PSI5:SEARCH:SETUP:ERROR:
SBOundary?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
ERROR:SBOundary 1

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:FRAMEinslot?

Function Queries all PSI5 signal slot specification search settings.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETUP]:
FRAMEinslot?
<x> = 1 to 4

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:FRAMEinslot:SNUMber

Function Sets or queries the slot number of the PSI5 signal slot specification search.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETUP]:
FRAMEinslot:SNUMber {<Nrf>}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
FRAMEinslot:SNUMber?
<x> = 1 to 4
<Nrf> = 1 to 6

Example :SERIALBUS1:PSI5:SEARCH:SETUP:
FRAMEINSLOT:SNUMBER 1
:SERIALBUS1:PSI5:SEARCH:SETUP:
FRAMEINSLOT:SNUMBER?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
FRAMEINSLOT:SNUMBER 1

:SERIALbus<x>:PSI5:SEARCH[:SETUP]:MODE

Function Sets or queries the PSI5 signal search mode setting.

Syntax :SERIALbus<x>:PSI5:SEARCH[:SETUP]:
MODE {DATA|ERROR|FRAMEinslot|SBIT|
SYNC}
:SERIALbus<x>:PSI5:SEARCH[:SETUP]:
MODE?
<x> = 1 to 4

Example :SERIALBUS1:PSI5:SEARCH:SETUP:
MODE DATA
:SERIALBUS1:PSI5:SEARCH:SETUP:MODE?
-> :SERIALBUS1:PSI5:SEARCH:SETUP:
MODE DATA

:SERIALbus<x>:PSI5:TREND<y>?

Function Queries all PSI5 signal analysis trend display settings.

Syntax :SERIALbus<x>:PSI5:TREND<y>?
<x> = 1 to 4
<y> = 1 to 4

:SERIALbus<x>:PSI5:TREND<y>:AScale

Function Executes auto scaling of the PSI5 signal analysis trend display.

Syntax :SERIALbus<x>:PSI5:TREND<y>:AScale
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:PSI5:TREND1:ASCALE

5.24 SERIALbus group

:SERIALbus<x>:PSI5:TREND<y>:CURSOR?

Function Queries all PSI5 signal analysis trend display cursor measurement settings.

Syntax :SERIALbus<x>:PSI5:TREND<y>:CURSOR?
<x> = 1 to 4
<y> = 1 to 4

:SERIALbus<x>:PSI5:TREND<y>:CURSOR:DT:VALUE?

Function Queries the time value between cursors on the PSI5 signal analysis trend display.

Syntax :SERIALbus<x>:PSI5:TREND<y>:CURSOR:
DT:VALUE?
<x> = 1 to 4
<y> = 1 to 4

:SERIALbus<x>:PSI5:TREND<y>:CURSOR:DV:VALUE?

Function Queries the vertical value between cursors on the PSI5 signal analysis trend display.

Syntax :SERIALbus<x>:PSI5:TREND<y>:CURSOR:
DV:VALUE?
<x> = 1 to 4
<y> = 1 to 4

:SERIALbus<x>:PSI5:TREND<y>:CURSOR:MODE

Function Sets or queries the cursor mode on the PSI5 signal analysis trend display.

Syntax :SERIALbus<x>:PSI5:TREND<y>:CURSOR:
MODE {<Boolean>}
:SERIALbus<x>:PSI5:TREND<y>:CURSOR:
MODE?
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:PSI5:TREND1:Cursors:
MODE ON
:SERIALBUS1:PSI5:TREND1:Cursors:MODE?
-> :SERIALBUS1:PSI5:TREND1:Cursors:
MODE 1

:SERIALbus<x>:PSI5:TREND<y>:CURSOR:POSITION<z>

Function Sets or queries the position of the specified cursor on the PSI5 signal analysis trend display.

Syntax :SERIALbus<x>:PSI5:TREND<y>:CURSOR:
POSITION<z> {<Nrf>}
:SERIALbus<x>:PSI5:TREND<y>:CURSOR:
POSITION<z>?
<x> = 1 to 4
<y> = 1 to 4
<z> = 1 to 2
<Nrf> = -5 to 5 (div), 10 div/display record length steps

Example :SERIALBUS1:PSI5:TREND1:Cursors:
POSITION1 -5
:SERIALBUS1:PSI5:TREND1:Cursors:
POSITION1?
-> :SERIALBUS1:PSI5:TREND1:Cursors:
POSITION1 -5

:SERIALbus<x>:PSI5:TREND<y>:CURSOR:T<z>:VALUE?

Function Queries the time value at the specified cursor on the PSI5 signal analysis trend display.

Syntax :SERIALbus<x>:PSI5:TREND<y>:CURSOR:
T<z>:VALUE?
<x> = 1 to 4
<y> = 1 to 4
<z> = 1 to 2

:SERIALbus<x>:PSI5:TREND<y>:CURSOR:V<z>:VALUE?

Function Queries the vertical value at the specified cursor on the PSI5 signal analysis trend display.

Syntax :SERIALbus<x>:PSI5:TREND<y>:CURSOR:
V<z>:VALUE?
<x> = 1 to 4
<y> = 1 to 4
<z> = 1 to 2

:SERIALbus<x>:PSI5:TREND<y>:DISPLAY

Function Sets or queries the on/off state of the PSI5 signal analysis trend display.

Syntax :SERIALbus<x>:PSI5:TREND<y>:
DISPLAY {<Boolean>}
:SERIALbus<x>:PSI5:TREND<y>:DISPLAY?
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:PSI5:TREND1:DISPLAY ON
:SERIALBUS1:PSI5:TREND1:DISPLAY?
-> :SERIALBUS1:PSI5:TREND1:DISPLAY 1

:SERIALbus<x>:PSI5:TREND<y>:HRANGE

Function Sets or queries the PSI5 signal analysis trend display source window.

Syntax :SERIALbus<x>:PSI5:TREND<y>:
HRANGE {MAIN|Z1|Z2}
:SERIALbus<x>:PSI5:TREND<y>:HRANGE?
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:PSI5:TREND1:HRANGE MAIN
:SERIALBUS1:PSI5:TREND1:HRANGE?
-> :SERIALBUS1:PSI5:TREND1:
HRANGE MAIN

:SERIALbus<x>:PSI5:TREND<y>:SOURCE

Function Sets or queries the target slot number of the PSI5 signal analysis trend display.

Syntax :SERIALbus<x>:PSI5:TREND<y>:
SOURCE {<NRF>}
:SERIALbus<x>:PSI5:TREND<y>:SOURCE?
<x> = 1 to 4
<y> = 1 to 4
<NRF> = 1 to 6

Example :SERIALBUS1:PSI5:TREND1:SOURCE 1
:SERIALBUS1:PSI5:TREND1:SOURCE?
-> :SERIALBUS1:PSI5:TREND1:SOURCE 1

:SERIALbus<x>:PSI5:TREND<y>:VERTICAL

Function Sets or queries the vertical range of the PSI5 signal analysis trend display.

Syntax :SERIALbus<x>:PSI5:TREND<y>:
VERTICAL {<NRF>,<NRF>}
:SERIALbus<x>:PSI5:TREND<y>:
VERTICAL?
<x> = 1 to 4
<y> = 1 to 4
<NRF> = -1.0000E+31 to 1.0000E+31

Example :SERIALBUS1:PSI5:TREND1:VERTICAL 1,
-1
:SERIALBUS1:PSI5:TREND1:VERTICAL?
-> :SERIALBUS1:PSI5:TREND1:
VERTICAL 1.0000000E+00,
-1.0000000E+00

:SERIALbus<x>:PSI5:TREND<y>:VTDisplay

Function Sets or queries the on/off status of the VT waveform display on the PSI5 signal analysis trend display.

Syntax :SERIALbus<x>:PSI5:TREND<y>:
VTDisplay {<Boolean>}
:SERIALbus<x>:PSI5:TREND<y>:
VTDisplay?
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:PSI5:TREND1:VTDISPLAY ON
:SERIALBUS1:PSI5:TREND1:VTDISPLAY?
-> :SERIALBUS1:PSI5:TREND1:
VTDISPLAY 1

:SERIALbus<x>:RWINDOW

Function Sets or queries the zoom window for showing the detected area.

Syntax :SERIALbus<x>:RWINDOW {Z1|Z2}
:SERIALbus<x>:RWINDOW?
<x> = 1 to 4

Example :SERIALBUS1:RWINDOW Z1
:SERIALBUS1:RWINDOW?
-> :SERIALBUS1:RWINDOW Z1

:SERIALbus<x>:SENT?

Function Queries all SENT signal analysis and search settings.

Syntax :SERIALbus<x>:SENT?
<x> = 1 to 4

:SERIALbus<x>:SENT:ANALyze?

Function Queries all SENT signal analysis settings.

Syntax :SERIALbus<x>:SENT:ANALyze?
<x> = 1 to 4

:SERIALbus<x>:SENT[:ANALyze]:SETup?

Function Queries all bus setup settings for SENT signal analysis.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup?
<x> = 1 to 4

5.24 SERIALbus group

:SERIALbus<x>:SENT[:ANALyze]:SETup:DISPlay

Function Sets or queries the display mode for the SENT signal analysis result.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:DISPlay {BOTH|FAST|SLOW}
:SERIALbus<x>:SENT[:ANALyze]:SETup:DISPlay?
<x> = 1 to 4

Example :SERIALBUS1:SENT:ANALYZE:SETUP:DISPLAY BOTH
:SERIALBUS1:SENT:ANALYZE:SETUP:DISPLAY?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:DISPLAY BOTH

:SERIALbus<x>:SENT[:ANALyze]:SETup:FASST?

Function Queries all SENT signal analysis fast channel settings.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:FAST?
<x> = 1 to 4

:SERIALbus<x>:SENT[:ANALyze]:SETup:FASST:DTYPE

Function Sets or queries the SENT signal analysis fast channel data format.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:DTYPE {NIBBLE|USER}
:SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:DTYPE?
<x> = 1 to 4

Example :SERIALBUS1:SENT:ANALYZE:SETUP:FAST:DTYPE NIBBLE
:SERIALBUS1:SENT:ANALYZE:SETUP:FAST:DTYPE?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:FAST:DTYPE NIBBLE

:SERIALbus<x>:SENT[:ANALyze]:SETup:FASST:USETup?

Function Queries all settings related to user-defined data of the SENT signal analysis fast channel.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:USETup?
<x> = 1 to 4

:SERIALbus<x>:SENT[:ANALyze]:SETup:FASST:USETup:DATA<y>?

Function Queries all settings related to user-defined data of the SENT signal analysis fast channel.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:USETup:DATA<y>?
<x> = 1 to 4
<y> = 1 to 4

:SERIALbus<x>:SENT[:ANALyze]:SETup:FASST:USETup:DATA<y>:MODE

Function Sets or queries the enable/disable condition of the specified user-defined data of the SENT signal analysis fast channel.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:USETup:DATA<y>:MODE {<Boolean>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:USETup:DATA<y>:MODE?
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:SENT:ANALYZE:SETUP:FAST:USETUP:DATA1:MODE ON
:SERIALBUS1:SENT:ANALYZE:SETUP:FAST:USETUP:DATA1:MODE?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:FAST:USETUP:DATA1:MODE 1

:SERIALbus<x>:SENT[:ANALyze]:SETup:FASST:USETup:DATA<y>:ORDER

Function Sets or queries the byte order of the specified user-defined data of the SENT signal analysis fast channel.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:USETup:DATA<y>:ORDER {BIG|LITTLE}
:SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:USETup:DATA<y>:ORDER?
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:SENT:ANALYZE:SETUP:FAST:USETUP:DATA1:ORDER BIG
:SERIALBUS1:SENT:ANALYZE:SETUP:FAST:USETUP:DATA1:ORDER?
->:SERIALBUS1:SENT:ANALYZE:SETUP:FAST:USETUP:DATA1:ORDER BIG

:SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:USETup:DATA<y>:SIZE

Function Sets or queries the data size of the specified user-defined data of the SENT signal analysis fast channel.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:USETup:DATA<y>:SIZE {<NRf>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:USETup:DATA<y>:SIZE?
<x> = 1 to 4
<y> = 1 to 4
<NRf> = 0 to 24

Example :SERIALBUS1:SENT:ANALYZE:SETUP:FAST:
USETUP:DATA1:SIZE 24
:SERIALBUS1:SENT:ANALYZE:SETUP:FAST:
USETUP:DATA1:SIZE?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FAST:USETUP:DATA1:SIZE 24

:SERIALbus<x>:SENT[:ANALyze]:SETup:FAST:USETup:MULTIplexing

Function Sets or queries the multiplexing enable/disable condition of the specified user-defined data of the SENT signal analysis fast channel.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:USETup:MULTIplexing {<Boolean>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FAST:USETup:MULTIplexing?
<x> = 1 to 4

Example :SERIALBUS1:SENT:ANALYZE:SETUP:FAST:
USETUP:MULTIPLExING ON
:SERIALBUS1:SENT:ANALYZE:SETUP:FAST:
USETUP:MULTIPLExING?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FAST:USETUP:MULTIPLExING 1

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMAT:CEFactor:SCPulses

Function Queries all SENT signal analysis format settings.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMAT:
<x> = 1 to 4

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMAT:CEFactor:SCPulses:CRCType

Function Queries all SENT signal analysis error factor settings.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMAT:CEFactor:
<x> = 1 to 4

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMAT:CEFactor:SCPulses:CRCType:LEGACY

Function Queries all status and communication error factor settings of SENT signal analysis.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMAT:CEFactor:
<x> = 1 to 4

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMAT:CEFactor:SCPulses:CRCType:LEGACY:BIT<y>?

Function Sets or queries status and communication error factor bit of SENT signal analysis.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMAT:CEFactor:BIT<y>?
<x> = 1 to 4
<y> = 0 to 1

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMAT:CEFactor:SCPulses:CRCType:LEGACY:BIT<y>:SCPulses

Function Sets or queries the consecutive calibration pulse error factor of SENT signal analysis.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMAT:CEFactor:SCPulses {OFF|OPT2|
POPTION}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMAT:CEFactor:SCPulses?
<x> = 1 to 4

Example :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CEFACTOR:SCPULSES OFF
:SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CEFACTOR:SCPULSES?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CEFACTOR:SCPULSES OFF

:SERIALbus<x>:SENT[:ANALyze]:SETup:FORMAT:CEFactor:SCPulses:CRCType:LEGACY:BIT<y>:SCPulses:CRCType

Function Sets or queries the SENT signal analysis CRC computation type.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMAT:CRCType {LEGACY|RECOMMENDED}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMAT:CRCType?
<x> = 1 to 4

Example :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CRCTYPE LEGACY
:SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CRCTYPE?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CRCTYPE LEGACY

5.24 SERIALbus group

:SERIALbus<x>:SENT[:ANALyze]:SETup:FOR Mat:CTICK

Function Sets or queries the SENT signal analysis clock tick value.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:CTICK {<NRf>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:CTICK?
<x> = 1 to 4
<NRf> = 1.00 μ s to 100.00 μ s, resolution: 0.01 μ s

Example :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CTICK 0.000001
:SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CTICK?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CTICK 1.000000E-06

:SERIALbus<x>:SENT[:ANALyze]:SETup:FOR Mat:CTOLerance

Function Sets or queries the SENT signal analysis clock tolerance.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:CTOLerance {<NRf>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:CTOLerance?
<x> = 1 to 4
<NRf> = $\pm$ 1.0% to $\pm$ 30.0%, resolution: 0.1%

Example :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CTOLerance 25.0
:SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CTOLerance?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:CTOLerance 25.0E+00

:SERIALbus<x>:SENT[:ANALyze]:SETup:FOR Mat:DNIBbles

Function Sets or queries the number of data nibbles for SENT signal analysis.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:DNIBbles {<NRf>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:DNIBbles?
<x> = 1 to 4
<NRf> = 1 to 6

Example :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:DNIBBLES 6
:SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:DNIBBLES?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:DNIBBLES 6

:SERIALbus<x>:SENT[:ANALyze]:SETup:FOR Mat:PPULse

Function Sets or queries the presence or absence of pause pulses for SENT signal analysis.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:PPULse {<Boolean>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:PPULse?
<x> = 1 to 4

Example :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:PPULSE ON
:SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:PPULSE?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:PPULSE 1

:SERIALbus<x>:SENT[:ANALyze]:SETup:FOR Mat:VERsion

Function Sets or queries the SENT signal analysis specification version.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:VERsion {APR2016|FEB2008|
JAN2010}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
FORMat:VERsion?
<x> = 1 to 4

Example :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:VERSION JAN2010
:SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:VERSION?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
FORMAT:VERSION JAN2010

:SERIALbus<x>:SENT[:ANALyze]:SETup:STY Pe

Function Sets or queries the SENT signal analysis slow channel format.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
STYPe {ENHanced|SHORT}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
STYPe?
<x> = 1 to 4

Example :SERIALBUS1:SENT:ANALYZE:SETUP:
STYPE ENHANCED
:SERIALBUS1:SENT:ANALYZE:SETUP:
STYPE?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
STYPE ENHANCED

**:SERIALbus<x>:SENT[:ANALyze]:SETup:SOU
Rce**

Function Sets or queries the SENT signal analysis source.

Syntax :SERIALbus<x>:SENT[:ANALyze]:SETup:
SOURce {<NRf>|BIT<y>|MATH<y>}
:SERIALbus<x>:SENT[:ANALyze]:SETup:
SOURce?
<x> = 1 to 4
<NRf> = 1 to 8
BIT<y>'s <y> = 1 to 8
MATH<y>'s <y> = 1 to 4

Example :SERIALBUS1:SENT:ANALYZE:SETUP:
SOURCE 1
:SERIALBUS1:SENT:ANALYZE:SETUP:
SOURCE?
-> :SERIALBUS1:SENT:ANALYZE:SETUP:
SOURCE 1

:SERIALbus<x>:SENT:DETail?

Function Queries all SENT signal analysis list settings.

Syntax :SERIALbus<x>:SENT:DETail?
<x> = 1 to 4

:SERIALbus<x>:SENT:DETail:DISPlay

Function Sets or queries the display mode for the SENT signal analysis list.

Syntax :SERIALbus<x>:SENT:DETail:
DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:SENT:DETail:DISPlay?
<x> = 1 to 4

Example :SERIALBUS1:SENT:DETAIL:DISPLAY FULL
:SERIALBUS1:SENT:DETAIL:DISPLAY?
-> :SERIALBUS1:SENT:DETAIL:
DISPLAY FULL

:SERIALbus<x>:SENT:DETail:LIST:ALL?

Function Queries all the data for all analysis numbers in the SENT signal analysis result list.

Syntax :SERIALbus<x>:SENT:DETail:LIST:ALL?
<x> = 1 to 4

Example :SERIALBUS1:SENT:DETAIL:LIST:ALL?
-> #8 (8-digit number of bytes)(data
sequence)

Description All the data for all analysis numbers are returned in <block data> format.

The data of each analysis number is separated by ASCII code 0AH.

:SERIALbus<x>:SENT:DETail:LIST:ITEM?

Function Queries all items that will be displayed in the SENT signal analysis result list.

Syntax :SERIALbus<x>:SENT:DETail:LIST:ITEM?
<x> = 1 to 4

Example :SERIALBUS1:SENT:DETAIL:LIST:ITEM?
-> :SERIALBUS1:SENT:DETAIL:LIST:
ITEM "No.,Time(ms),Sync(us),
Tick(us),S&C,Data,CRC,Length(tick),
Information,SlowCH,"

Description The items displayed in the analysis result list varies depending on the :SERIALbus<x>:SENT[:ANALyze]:SETup:DISPlay {BOTH|FAST|SLOW} setting.

:SERIALbus<x>:SENT:DETail:LIST:VALue?

Function Queries all the data for the specified analysis number in the SENT signal analysis result list.

Syntax :SERIALbus<x>:SENT:DETail:LIST:
VALue? {<NRf>|MAXimum|MINimum}
<x> = 1 to 4
<NRf> = -100000 to 100000

Example :SERIALBUS1:SENT:DETAIL:LIST:
VALUE? 0
-> :SERIALBUS1:SENT:DETAIL:LIST:
VALUE "0,-0.1672,168.00,3.00,1100,6,
3,5,E,B,9,3,283.73,,,"

Description The items displayed in the analysis result list varies depending on the :SERIALbus<x>:SENT[:ANALyze]:SETup:DISPlay {BOTH|FAST|SLOW} setting.

:SERIALbus<x>:SENT:SEARCh?

Function Queries all SENT signal search settings.

Syntax :SERIALbus<x>:SENT:SEARCh?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCh:ABORT

Function Aborts the SENT signal search.

Syntax :SERIALbus<x>:SENT:SEARCh:ABORT
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:ABORT

:SERIALbus<x>:SENT:SEARCh:EXECute

Function Executes a SENT signal search.

Syntax :SERIALbus<x>:SENT:SEARCh:EXECute
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:EXECUTE

5.24 SERIALbus group

:SERIALbus<x>:SENT:SEARCH:SELEct

Function Sets which detected point to display in the SENT signal search zoom window and queries the zoom position of the detected point.

Syntax :SERIALbus<x>:SENT:SEARCH:
SELEct {<NRf>|MAXimum}
:SERIALbus<x>:SENT:SEARCH:SELEct?
<x> = 1 to 4
<NRf> = 0 to 49999

Example :SERIALBUS1:SENT:SEARCH:SELECT 1
:SERIALBUS1:SENT:SEARCH:SELECT?
-> :SERIALBUS1:SENT:SEARCH:
SELECT 1.50000000

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:SENT:SEARCH:SELEct? MAXimum

Function Queries the number of detected points in the SENT signal search.

Syntax :SERIALbus<x>:SENT:SEARCH:
SELEct? {MAXimum}
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:
SELECT? MAXimum
-> :SERIALBUS1:SENT:SEARCH:
SELECT 100

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:SENT:SEARCH:SETup?

Function Queries all SENT signal search setup settings.

Syntax :SERIALbus<x>:SENT:SEARCH:SETup?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETup]:ERRor?

Function Queries all SENT signal search error settings.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
ERRor?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETup]:ERRor:FCRC

Function Sets or queries the SENT signal search fast channel CRC error.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
ERRor:FCRC {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
ERRor:FCRC?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
FCRC ON
:SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
FCRC?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
ERROR:FCRC 1

:SERIALbus<x>:SENT:SEARCH[:SETup]:ERRor:NDValue

Function Sets or queries the SENT signal search nibble data value error.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
ERRor:NDValue {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
ERRor:NDValue?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
NDVALUE ON
:SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
NDVALUE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
ERROR:NDVALUE 1

:SERIALbus<x>:SENT:SEARCH[:SETup]:ERRor:NNUmber

Function Sets or queries the SENT signal search nibble data count error.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
ERRor:NNUmber {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
ERRor:NNUmber?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
NNUmber ON
:SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
NNUmber?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
ERROR:NNUmber 1

:SERIALbus<x>:SENT:SEARCH[:SETup]:ERROR:SAComm

Function Sets or queries the SENT signal search status and communication error.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
ERROR:SAComm {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
ERROR:SAComm?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
SACOMM ON
:SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
SACOMM?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
ERROR:SACOMM 1

:SERIALbus<x>:SENT:SEARCH[:SETup]:ERROR:SCPulses

Function Sets or queries the consecutive calibration pulse error for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
ERROR:SCPulses{<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
ERROR:SCPulses?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
SCPULSESON
:SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
SCPULSES?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
ERROR:SCPULSES 1

Description When :SERIALbus<x>:SENT[:ANALyze]:SETup:FORMAT:CEFactor:SCPulses is set to OFF, this setting is fixed to OFF.

:SERIALbus<x>:SENT:SEARCH[:SETup]:ERROR:SCRC

Function Sets or queries the SENT signal search slow channel CRC error.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
ERROR:SCRC {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
ERROR:SCRC?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
SCRC ON
:SERIALBUS1:SENT:SEARCH:SETUP:ERROR:
SCRC?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
ERROR:SCRC 1

:SERIALbus<x>:SENT:SEARCH[:SETup]:FDATa?

Function Queries all fast channel data settings for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
FDATa?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETup]:FDATa:DATA<y>?

Function Queries all settings related to the specified fast channel user data for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
FDATa:DATA<y>?
<x> = 1 to 4
<y> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETup]:FDATa:DATA<y>:CONDition

Function Sets or queries the comparison condition of the specified user data of the SENT signal search fast channel.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
FDATa:DATA<y>:CONDition {EQUAL|
GREATER|INSide|LESS|NOTEQUAL|
OUTSide}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
FDATa:DATA<y>:CONDition?
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DATA1:CONDITION EQUAL
:SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DATA1:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
FDATA:DATA1:CONDITION EQUAL

:SERIALbus<x>:SENT:SEARCH[:SETup]:FDATa:DATA<y>:DECimal<z>

Function Sets or queries the data of the specified user data of the SENT signal search fast channel in decimal notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
FDATa:DATA<y>:DECimal<z> {<NRF>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
FDATa:DATA<y>:DECimal<z>?
<x> = 1 to 4
<y> = 1 to 4
<z> = 1 to 2

<NRF> = See the Features Guide for this information.

Example :SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DATA1:DECIMAL1 1
:SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DATA1:DECIMAL1?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
FDATA:DATA1:DECIMAL1 1

5.24 SERIALbus group

:SERIALbus<x>:SENT:SEARCH[:SETup]:FDATA:DATA<y>:MODE

Function Sets or queries the data enable/disable condition of the specified user data of the SENT signal search fast channel.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
FDATA:DATA<y>:MODE {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
FDATA:DATA<y>:MODE?
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DATA1:MODE ON
:SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DATA1:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
FDATA:DATA1:MODE 1

:SERIALbus<x>:SENT:SEARCH[:SETup]:FDATA:DNIBbles?

Function Queries all fast channel nibble data settings for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
FDATA:DNIBbles?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETup]:FDATA:DNIBbles:CONDition

Function Sets or queries the nibble data comparison condition of the SENT signal search fast channel.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
FDATA:DNIBbles:CONDition {FALSE|
TRUE}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
FDATA:DNIBbles:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DNIBBLES:CONDITION TRUE
:SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DNIBBLES:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
FDATA:DNIBBLES:CONDITION TRUE

:SERIALbus<x>:SENT:SEARCH[:SETup]:FDATA:DNIBbles:HEXa

Function Sets the SENT signal search fast channel nibble data in hexadecimal notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
FDATA:DNIBbles:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DNIBBLES:HEXa "112233"

:SERIALbus<x>:SENT:SEARCH[:SETup]:FDATA:DNIBbles:PATtern

Function Sets or queries the SENT signal search fast channel nibble data in binary notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
FDATA:DNIBbles:PATtern {<String>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
FDATA:DNIBbles:PATtern?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DNIBBLES:
PATTERN "111100001111000011110000"
:SERIALBUS1:SENT:SEARCH:SETUP:FDATA:
DNIBBLES:PATTERN?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
FDATA:DNIBBLES:
PATTERN "111100001111000011110000"

:SERIALbus<x>:SENT:SEARCH[:SETup]:FSAComm?

Function Queries all fast channel status and communication nibble settings for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
FSAComm?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETup]:FSAComm:HEXa

Function Sets the SENT signal search fast channel status and communication nibble data in hexadecimal notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
FSAComm:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:
FSACOMM:HEXa "F"

:SERIALbus<x>:SENT:SEARCH[:SETup]:FSAComm:PATtern

Function Sets or queries the SENT signal search fast channel status and communication nibble data in binary notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
FSAComm:PATtern {<String>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
FSAComm:PATtern?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:
FSACOMM:PATTERN "1111"
:SERIALBUS1:SENT:SEARCH:SETUP:
FSACOMM:PATTERN?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
FSACOMM:PATTERN "1111"

:SERIALbus<x>:SENT:SEARCH[:SETUP]:MODE

Function Sets or queries the SENT signal search mode.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
MODE {EFAST|ERROR|ESLOW|FDATA|
FSACOMM|SDATA}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
MODE?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:
MODE ERROR
:SERIALBUS1:SENT:SEARCH:SETUP:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
MODE ERROR

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA?

Function Queries all slow channel data settings for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED?

Function Queries all slow channel enhanced type settings for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:CBIT

Function Sets or queries the SENT signal search slow channel enhanced type configuration bit.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:CBIT {D12BIT|D16BIT}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:CBIT?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:CBIT D12BIT
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:CBIT?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:CBIT D12BIT

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D12BIT?

Function Queries all slow channel enhanced type 12-bit data and 8-bit ID settings for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D12BIT?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D12BIT:DATA?

Function Queries all slow channel enhanced type 12-bit data and 8-bit ID data settings for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D12BIT:DATA?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D12BIT:DATA:CONDITION

Function Sets or queries the data comparison condition of slow channel enhanced type 12-bit data and 8-bit ID for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D12BIT:DATA:
CONDITION {BETWEEN|EQUAL|FALSE|
GREATER|LESS|NOTBETWEEN|NOTEQUAL|
TRUE}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D12BIT:DATA:
CONDITION?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:CONDITION FALSE
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D12BIT:DATA:
CONDITION FALSE

5.24 SERIALbus group

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATa:ENHanced:D12Bit:DATA:DECimal<y>

Function Sets or queries the slow channel enhanced type 12-bit data and 8-bit ID data for SENT signal searching in decimal notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:ENHanced:D12Bit:DATA:
DECimal<y> {<NRF>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:ENHanced:D12Bit:DATA:
DECimal<y>?
<x> = 1 to 4
<y> = 1 to 2
<NRF> = See the Features Guide for this information.

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:DECIMAL1 123
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:DECIMAL1?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D12BIT:DATA:
DECIMAL1 123

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATa:ENHanced:D12Bit:DATA:HEXa

Function Sets the slow channel enhanced type 12-bit data and 8-bit ID data for SENT signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:ENHanced:D12Bit:DATA:
HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:HEXA "123"

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATa:ENHanced:D12Bit:DATA:MODE

Function Sets or queries the data enable/disable condition of the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:ENHanced:D12Bit:DATA:
MODE {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:ENHanced:D12Bit:DATA:MODE?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:MODE ON
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D12BIT:DATA:MODE 1

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATa:ENHanced:D12Bit:DATA:PATtern

Function Sets or queries the slow channel enhanced type 12-bit data and 8-bit ID data for SENT signal searching in binary notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:ENHanced:D12Bit:DATA:
PATtern {<String>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:ENHanced:D12Bit:DATA:PATtern?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:
PATTERN "111101010101"
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:DATA:PATTERN?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D12BIT:DATA:
PATTERN "111101010101"

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATa:ENHanced:D12Bit:ID?

Function Queries all ID settings related to the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:ENHanced:D12Bit:ID?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATa:ENHanced:D12Bit:ID:CONDition

Function Sets or queries the ID comparison condition of slow channel enhanced type 12-bit data and 8-bit ID for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:ENHanced:D12Bit:ID:
CONDition {EQUAL|GREATER|INSIDE|
LESS|NOTEQUAL|OUTSIDE}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:ENHanced:D12Bit:ID:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:ID:CONDITION EQUAL
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:ID:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D12BIT:ID:
CONDITION EQUAL

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D12BIT:ID:DECIMAL<y>

Function Sets or queries the ID of the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal searching in decimal notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D12BIT:ID:
DECIMAL<y> {<NRf>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D12BIT:ID:DECIMAL<y>?
<x> = 1 to 4
<y> = 1 to 2
<NRf> = See the Features Guide for this information.

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:ID:DECIMAL1 0
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:ID:DECIMAL1?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D12BIT:ID:DECIMAL1 0

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D12BIT:ID:MODE

Function Sets or queries the ID enable/disable condition of the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D12BIT:ID:
MODE {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D12BIT:ID:MODE?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:ID:MODE ON
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D12BIT:ID:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D12BIT:ID:MODE 1

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D16BIT?

Function Queries all slow channel enhanced type 16-bit data and 4-bit ID settings for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D16BIT:DATA?

Function Queries all slow channel enhanced type 16-bit data and 4-bit ID data settings for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:DATA?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D16BIT:DATA:CONDITION

Function Sets or queries the data comparison condition of slow channel enhanced type 16-bit data and 4-bit ID for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:DATA:
CONDITION {BETWEEN|EQUAL|FALSE|
GREATER|LESS|NOTBETWEEN|NOTEQUAL|
TRUE}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:DATA:
CONDITION?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:DATA:CONDITION FALSE
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:DATA:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D16BIT:DATA:
CONDITION FALSE

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D16BIT:DATA:DECIMAL<y>

Function Sets or queries the slow channel enhanced type 16-bit data and 4-bit ID data for SENT signal searching in decimal notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:DATA:
DECIMAL<y> {<NRf>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:DATA:
DECIMAL<y>?
<x> = 1 to 4
<y> = 1 to 2
<NRf> = See the Features Guide for this information.

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:DATA:DECIMAL1 123
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:DATA:DECIMAL1?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D16BIT:DATA:
DECIMAL1 123

5.24 SERIALbus group

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D16BIT:DATA:HEXA

Function Sets the slow channel enhanced type 16-bit data and 4-bit ID data for SENT signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:DATA:
HEXA {<String>}
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:DATA:HEXA "0000"

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D16BIT:DATA:MODE

Function Sets or queries the data enable/disable condition of the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:DATA:
MODE {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:DATA:MODE?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:DATA:MODE ON
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:DATA:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D16BIT:DATA:MODE 1

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D16BIT:DATA:PATTERN

Function Sets or queries the slow channel enhanced type 16-bit data and 4-bit ID data for SENT signal searching in binary notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:DATA:
PATTERN {<String>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:DATA:PATTERN?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:DATA:
PATTERN "1111000011110000"
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:DATA:PATTERN?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D16BIT:DATA:
PATTERN "1111000011110000"

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D16BIT:ID?

Function Queries all ID settings related to the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:ID?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D16BIT:ID:CONDITION

Function Sets or queries the ID comparison condition of slow channel enhanced type 16-bit data and 4-bit ID for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:ID:
CONDITION {EQUAL|GREATER|INSIDE|
LESS|NOTEQUAL|OUTSIDE}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:ID:CONDITION?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:ID:CONDITION EQUAL
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:ID:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D16BIT:ID:
CONDITION EQUAL

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:ENHANCED:D16BIT:ID:DECIMAL<y>

Function Sets or queries the ID of the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal searching in decimal notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:ID:
DECIMAL<y> {<NRF>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:ENHANCED:D16BIT:ID:DECIMAL<y>?
<x> = 1 to 4
<y> = 1 to 2
<NRF> = See the Features Guide for this information.

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:ID:DECIMAL1 0
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
ENHANCED:D16BIT:ID:DECIMAL1?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:ENHANCED:D16BIT:ID:DECIMAL1 0

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATa:ENHanced:D16Bit:ID:MODE

Function Sets or queries the ID enable/disable condition of the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:ENHanced:D16Bit:ID:
MODE {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:ENHanced:D16Bit:ID:MODE?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATa:
ENHANCED:D16BIT:ID:MODE ON
:SERIALBUS1:SENT:SEARCH:SETUP:SDATa:
ENHANCED:D16BIT:ID:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATa:ENHANCED:D16BIT:ID:MODE 1

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATa:SHORT?

Function Queries all slow channel short type settings for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:SHORT?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATa:SHORT:DATA?

Function Queries all slow channel short type data settings for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:SHORT:DATA?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATa:SHORT:DATA:CONDition

Function Sets or queries the short type data comparison condition of the SENT signal search slow channel.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:SHORT:DATA:CONDition {EQUAL|
FALSE|GREATER|INSIDE|LESS|NOTEQUAL|
OUTSIDE|TRUE}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:SHORT:DATA:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATa:
SHORT:DATA:CONDITION EQUAL
:SERIALBUS1:SENT:SEARCH:SETUP:SDATa:
SHORT:DATA:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATa:SHORT:DATA:CONDITION EQUAL

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATa:SHORT:DATA:DECimal<y>

Function Sets or queries the SENT signal search slow channel short type data in decimal notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:SHORT:DATA:DECimal<y> {<Nrf>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:SHORT:DATA:DECimal<y>?
<x> = 1 to 4
<y> = 1 to 2
<Nrf> = See the Features Guide for this information.

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATa:
SHORT:DATA:DECIMAL1 123
:SERIALBUS1:SENT:SEARCH:SETUP:SDATa:
SHORT:DATA:DECIMAL1?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATa:SHORT:DATA:DECIMAL1 123

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATa:SHORT:DATA:HEXa

Function Sets the SENT signal search slow channel short type data in hexadecimal notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:SHORT:DATA:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATa:
SHORT:DATA:HEXa "00"

:SERIALbus<x>:SENT:SEARCH[:SETup]:SDATa:SHORT:DATA:MODE

Function Sets or queries the short type data enable/disable condition of the SENT signal search slow channel.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:SHORT:DATA:MODE {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETup]:
SDATa:SHORT:DATA:MODE?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATa:
SHORT:DATA:MODE ON
:SERIALBUS1:SENT:SEARCH:SETUP:SDATa:
SHORT:DATA:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATa:SHORT:DATA:MODE 1

5.24 SERIALbus group

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:DATA:PATTERN

Function Sets or queries the SENT signal search slow channel short type data in binary notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:DATA:PATTERN {<String>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:DATA:PATTERN?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:DATA:PATTERN "11110000"
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:DATA:PATTERN?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:SHORT:DATA:PATTERN "11110000"

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID?

Function Queries all slow channel short type ID settings for SENT signal searching.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:ID?
<x> = 1 to 4

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID:CONDition

Function Sets or queries the short type ID comparison condition of the SENT signal search slow channel.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:ID:CONDition {EQUAL|
GREATER|INSIDE|LESS|NOTEQUAL|
OUTSIDE}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:ID:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:ID:CONDITION EQUAL
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:ID:CONDITION?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:SHORT:ID:CONDITION EQUAL

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID:DECimal<y>

Function Sets or queries the SENT signal search slow channel short type ID in decimal notation.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:ID:DECimal<y> {<NRF>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:ID:DECimal<y>?
<x> = 1 to 4
<y> = 1 to 2
<NRF> = See the Features Guide for this information.

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:ID:DECIMAL 0
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:ID:DECIMAL?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:SHORT:ID:DECIMAL 0

:SERIALbus<x>:SENT:SEARCH[:SETUP]:SDATA:SHORT:ID:MODE

Function Sets or queries the short type ID enable/disable condition of the SENT signal search slow channel.

Syntax :SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:ID:MODE {<Boolean>}
:SERIALbus<x>:SENT:SEARCH[:SETUP]:
SDATA:SHORT:ID:MODE?
<x> = 1 to 4

Example :SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:ID:MODE ON
:SERIALBUS1:SENT:SEARCH:SETUP:SDATA:
SHORT:ID:MODE?
-> :SERIALBUS1:SENT:SEARCH:SETUP:
SDATA:SHORT:ID:MODE 1

:SERIALbus<x>:SENT:TREND<y>?

Function Queries all SENT signal analysis trend display settings.

Syntax :SERIALbus<x>:SENT:TREND<y>?
<x> = 1 to 4
<y> = 1 to 4

:SERIALbus<x>:SENT:TREND<y>:ASCale

Function Executes auto scaling of the SENT signal analysis trend display.

Syntax :SERIALbus<x>:SENT:TREND<y>:ASCale
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:SENT:TREND1:ASCALe

:SERIALbus<x>:SENT:TREND<y>:CURSOR?

Function Queries all SENT signal analysis trend display cursor measurement settings.

Syntax :SERIALbus<x>:SENT:TREND<y>:CURSOR?
 <x> = 1 to 4
 <y> = 1 to 4

:SERIALbus<x>:SENT:TREND<y>:CURSOR:DT:VALUE?

Function Queries the time value between cursors on the SENT signal analysis trend display.

Syntax :SERIALbus<x>:SENT:TREND<y>:CURSOR:
 DT:VALUE?
 <x> = 1 to 4
 <y> = 1 to 4

:SERIALbus<x>:SENT:TREND<y>:CURSOR:DV:VALUE?

Function Queries the vertical value between cursors on the SENT signal analysis trend display.

Syntax :SERIALbus<x>:SENT:TREND<y>:CURSOR:
 DV:VALUE?
 <x> = 1 to 4
 <y> = 1 to 4

:SERIALbus<x>:SENT:TREND<y>:CURSOR:MODE

Function Sets or queries the automated measurement mode of the SENT signal analysis trend display.

Syntax :SERIALbus<x>:SENT:TREND<y>:CURSOR:
 MODE {<Boolean>}
 :SERIALbus<x>:SENT:TREND<y>:CURSOR:
 MODE?
 <x> = 1 to 4
 <y> = 1 to 4

Example :SERIALBUS1:SENT:TREND1:CURSOR:
 MODE ON
 :SERIALBUS1:SENT:TREND1:CURSOR:MODE?
 -> :SERIALBUS1:SENT:TREND1:CURSOR:
 MODE 1

:SERIALbus<x>:SENT:TREND<y>:CURSOR:POSITION<z>

Function Sets or queries the position of the specified cursor on the SENT signal analysis trend display.

Syntax :SERIALbus<x>:SENT:TREND<y>:CURSOR:
 POSITION<z> {<NRF>}
 :SERIALbus<x>:SENT:TREND<y>:CURSOR:
 POSITION<z>?
 <x> = 1 to 4
 <y> = 1 to 4
 <z> = 1 to 2
 <NRF> = -5 to 5 (div), 10 div/display record length steps

Example :SERIALBUS1:SENT:TREND1:CURSOR:
 POSITION1 2
 :SERIALBUS1:SENT:TREND1:CURSOR:
 POSITION1?
 -> :SERIALBUS1:SENT:TREND1:CURSOR:
 POSITION1 2.00E+00

:SERIALbus<x>:SENT:TREND<y>:CURSOR:T<z>:VALUE?

Function Queries the time value at the specified cursor on the SENT signal analysis trend display.

Syntax :SERIALbus<x>:SENT:TREND<y>:CURSOR:
 T<z>:VALUE?
 <x> = 1 to 4
 <y> = 1 to 4
 <z> = 1 to 2

:SERIALbus<x>:SENT:TREND<y>:CURSOR:V<z>:VALUE?

Function Queries the vertical value at the specified cursor on the SENT signal analysis trend display.

Syntax :SERIALbus<x>:SENT:TREND<y>:CURSOR:
 V<z>:VALUE?
 <x> = 1 to 4
 <y> = 1 to 4
 <z> = 1, 2

:SERIALbus<x>:SENT:TREND<y>:DISPLAY

Function Sets or queries whether to show (ON) or hide (OFF) the SENT signal analysis trend.

Syntax :SERIALbus<x>:SENT:TREND<y>:
 DISPLAY {<Boolean>}
 :SERIALbus<x>:SENT:TREND<y>:DISPLAY?
 <x> = 1 to 4
 <y> = 1 to 4

Example :SERIALBUS1:SENT:TREND1:DISPLAY ON
 :SERIALBUS1:SENT:TREND1:DISPLAY?
 -> :SERIALBUS1:SENT:TREND1:DISPLAY 1

5.24 SERIALbus group

:SERIALbus<x>:SENT:TREND<y>:FCONTROL

Function Sets or queries the frame control value of the SENT signal analysis trend display.

Syntax :SERIALbus<x>:SENT:TREND<y>:
FCONTROL {<String>}
:SERIALbus<x>:SENT:TREND<y>:
FCONTROL?
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:SENT:TREND1:FCONTROL "1"
:SERIALBUS1:SENT:TREND1:FCONTROL?
-> :SERIALBUS1:SENT:TREND1:
FCONTROL 1

:SERIALbus<x>:SENT:TREND<y>:HRANGE

Function Sets or queries the SENT signal analysis trend display source window.

Syntax :SERIALbus<x>:SENT:TREND<y>:
HRANGE {MAIN|Z1|Z2}
:SERIALbus<x>:SENT:TREND<y>:HRANGE?
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:SENT:TREND1:HRANGE MAIN
:SERIALBUS1:SENT:TREND1:HRANGE?
-> :SERIALBUS1:SENT:TREND1:
HRANGE MAIN

:SERIALbus<x>:SENT:TREND<y>:SID

Function Sets or queries the slow channel ID of the SENT signal analysis trend display in hexadecimal notation.

Syntax :SERIALbus<x>:SENT:TREND<y>:
SID {<String>}
:SERIALbus<x>:SENT:TREND<y>:SID?
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:SENT:TREND1:SID "01"
:SERIALBUS1:SENT:TREND1:SID?
-> :SERIALBUS1:SENT:TREND1:SID "01"

Description • If the slow channel type is set to Short, the value of the first digit of this command's parameter will be set.
• Set this command's parameter using a 2-digit hexadecimal regardless of the slow channel type setting.

:SERIALbus<x>:SENT:TREND<y>:SOURCE

Function Sets or queries the SENT signal analysis trend display source channel.

Syntax :SERIALbus<x>:SENT:TREND<y>:
SOURCE {FAST|SLOW}
:SERIALbus<x>:SENT:TREND<y>:SOURCE?
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:SENT:TREND1:SOURCE FAST
:SERIALBUS1:SENT:TREND1:SOURCE?
-> :SERIALBUS1:SENT:TREND1:
SOURCE FAST

:SERIALbus<x>:SENT:TREND<y>:UDATA

Function Sets or queries the fast channel user-defined data of the SENT signal analysis trend display.

Syntax :SERIALbus<x>:SENT:TREND<y>:
UDATA {<NRF>}
:SERIALbus<x>:SENT:TREND<y>:UDATA?
<x> = 1 to 4
<y> = 1 to 4
<NRF> = 1 to 4

Example :SERIALBUS1:SENT:TREND1:UDATA 1
:SERIALBUS1:SENT:TREND1:UDATA?
-> :SERIALBUS1:SENT:TREND1:UDATA 1

:SERIALbus<x>:SENT:TREND<y>:VERTICAL

Function Sets or queries the vertical range of the SENT signal analysis trend display.

Syntax :SERIALbus<x>:SENT:TREND<y>:
VERTICAL {<NRF>,<NRF>}
:SERIALbus<x>:SENT:TREND<y>:
VERTICAL?
<x> = 1 to 4
<y> = 1 to 4
<NRF> = -1.0000E+31 to 1.0000E+31

Example :SERIALBUS1:SENT:TREND1:VERTICAL 1,
-1
:SERIALBUS1:SENT:TREND1:VERTICAL?
-> :SERIALBUS1:SENT:TREND1:
VERTICAL 1.0000000E+00,
-1.0000000E+00

:SERIALbus<x>:SENT:TREND<y>:VTDISPLAY

Function Sets or queries the on/off status of the VT waveform display on the SENT signal analysis trend display.

Syntax :SERIALbus<x>:SENT:TREND<y>:
VTDISPLAY {<Boolean>}
:SERIALbus<x>:SENT:TREND<y>:
VTDISPLAY?
<x> = 1 to 4
<y> = 1 to 4

Example :SERIALBUS1:SENT:TREND1:VTDISPLAY ON
:SERIALBUS1:SENT:TREND1:VTDISPLAY?
-> :SERIALBUS1:SENT:TREND1:
VTDISPLAY 1

:SERIALbus<x>:SOURCE?

Function Queries all analysis and search settings.

Syntax :SERIALbus<x>:SOURCE?
<x> = 1 to 4

:SERIALbus<x>:SOURCE:{CHANNEL<y>|MATH<y>}?

Function Queries all source waveform settings.

Syntax :SERIALbus<x>:SOURCE:{CHANNEL<y>|
MATH<y>}?
<x> = 1 to 4
CHANNEL<y>'s <y> = 1 to 4
MATH<y>'s <y> = 1 to 4

:SERIALbus<x>:SOURCE:{CHANNEL<y>|MATH<y>}:HYSTERESIS

Function Sets or queries the source waveform hysteresis.

Syntax :SERIALbus<x>:SOURCE:{CHANNEL<y>|
MATH<y>}:HYSTERESIS {<NRf>}
:SERIALbus<x>:SOURCE:{CHANNEL<y>|
MATH<y>}:HYSTERESIS?
<x> = 1 to 4
CHANNEL<y>'s <y> = 1 to 4
MATH<y>'s <y> = 1 to 4
<NRf> = See the Features Guide for this information.

:SERIALbus<x>:SOURCE:{CHANNEL<y>|MATH<y>}:LEVEL

Function Sets or queries the source waveform level.

Syntax :SERIALbus<x>:SOURCE:{CHANNEL<y>|
MATH<y>}:LEVEL {<Voltage>,<Voltage>}
:SERIALbus<x>:SOURCE:{CHANNEL<y>|
MATH<y>}:LEVEL?
<x> = 1 to 4
CHANNEL<y>'s <y> = 1 to 4
MATH<y>'s <y> = 1 to 4
<Voltage> = See the Features Guide for this information.

Example :SERIALBUS1:SOURCE:CHANNEL1:LEVEL 0V
:SERIALBUS1:SOURCE:CHANNEL1:LEVEL?
-> :SERIALBUS1:SOURCE:CHANNEL1:
LEVEL 0.000E+00

:SERIALbus<x>:SPATTERN?

Function Queries all user-defined bus signal analysis and search settings.

Syntax :SERIALbus<x>:SPATTERN?
<x> = 1 to 4

:SERIALbus<x>:SPATTERN:ANALYZE?

Function Queries all user-defined bus signal analysis execution settings.

Syntax :SERIALbus<x>:SPATTERN:ANALYZE?
<x> = 1 to 4

:SERIALbus<x>:SPATTERN[:ANALYZE]:SETUP?

Function Queries all user-defined bus signal analysis settings.

Syntax :SERIALbus<x>:SPATTERN[:ANALYZE]:
SETUP?
<x> = 1 to 4

:SERIALbus<x>:SPATTERN[:ANALYZE]:SETUP:BRATE

Function Sets or queries the user-defined bus signal analysis bit rate.

Syntax :SERIALbus<x>:SPATTERN[:ANALYZE]:
SETUP:BRATE {<NRf>}
:SERIALbus<x>:SPATTERN[:ANALYZE]:
SETUP:BRATE?
<x> = 1 to 4
<NRf> = 1000 to 50000000(bps)

Example :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
BRATE 1000
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
BRATE?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:BRATE 1000

5.24 SERIALbus group

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:CLOCK?

Function Queries all user-defined bus signal analysis clock signal settings.

Syntax :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CLOCK?
<x> = 1 to 4

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:CLOCK:MODE

Function Sets or queries the enable/disable condition of the user-defined bus signal analysis clock signal.

Syntax :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CLOCK:MODE {<Boolean>}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CLOCK:MODE?
<x> = 1 to 4

Example :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CLOCK:MODE ON
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CLOCK:MODE?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:CLOCK:MODE 1

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:CLOCK:POLarity

Function Sets or queries the slope of the user-defined bus signal analysis clock signal.

Syntax :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CLOCK:POLarity {FALL|RISE}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CLOCK:POLarity?
<x> = 1 to 4

Example :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CLOCK:POLARITY FALL
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CLOCK:POLARITY?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:CLOCK:POLARITY FALL

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:CLOCK:SOURce

Function Sets or queries the user-defined bus signal analysis clock signal.

Syntax :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CLOCK:SOURce {<Nrf>}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CLOCK:SOURce?
<x> = 1 to 4
<Nrf> = 1 to 4

Example :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CLOCK:SOURCE 1
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CLOCK:SOURCE?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:CLOCK:SOURCE 1

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:CS?

Function Queries all user-defined bus signal analysis chip select signal settings.

Syntax :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CS?
<x> = 1 to 4

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:CS:ACTive

Function Sets or queries the active state of the user-defined bus signal analysis chip select signal.

Syntax :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CS:ACTive {HIGH|LOW}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CS:ACTive?
<x> = 1 to 4

Example :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CS:ACTIVE HIGH
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CS:ACTIVE?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:CS:ACTIVE HIGH

**:SERIALbus<x>:SPATtern[:ANALyze]:SETup
:CS:SOURce**

Function Sets or queries the user-defined bus signal analysis chip select signal.

Syntax :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CS:SOURce {<Nrf>|NONE}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:CS:SOURce?
<x> = 1 to 4
<Nrf> = 1 to 4

Example :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CS:SOURCE 1
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
CS:SOURCE?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:CS:SOURCE 1

**:SERIALbus<x>:SPATtern[:ANALyze]:SETup
:DATA?**

Function Queries all user-defined bus signal analysis data signal settings.

Syntax :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:DATA?
<x> = 1 to 4

**:SERIALbus<x>:SPATtern[:ANALyze]:SETup
:DATA:ACTive**

Function Sets or queries the active state of the user-defined bus signal analysis data signal.

Syntax :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:DATA:ACTive {HIGH|LOW}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:DATA:ACTive?
<x> = 1 to 4

Example :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
DATA:ACTIVE HIGH
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
DATA:ACTIVE?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:DATA:ACTIVE HIGH

**:SERIALbus<x>:SPATtern[:ANALyze]:SETup
:DATA:SOURce**

Function Sets or queries the user-defined bus signal analysis data signal.

Syntax :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:DATA:SOURce {<Nrf>}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:DATA:SOURce?
<x> = 1 to 4
<Nrf> = 1 to 4

Example :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
DATA:SOURCE 1
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
DATA:SOURCE?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:DATA:SOURCE 1

**:SERIALbus<x>:SPATtern[:ANALyze]:SETup
:LATCh?**

Function Queries all user-defined bus signal analysis latch signal settings.

Syntax :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:LATCh?
<x> = 1 to 4

**:SERIALbus<x>:SPATtern[:ANALyze]:SETup
:LATCh:POLarity**

Function Sets or queries the slope of the user-defined bus signal analysis latch signal.

Syntax :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:LATCh:POLarity {FALL|RISE}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:LATCh:POLarity?
<x> = 1 to 4

Example :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
LATCH:POLARITY FALL
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
LATCH:POLARITY?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:LATCH:POLARITY FALL

5.24 SERIALbus group

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:LATCh:SOURce

Function Sets or queries the user-defined bus signal analysis latch signal.

Syntax :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:LATCh:SOURce {<NRf>|NONE}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:LATCh:SOURce?
<x> = 1 to 4
<NRf> = 1 to 4

Example :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
LATCH:SOURCE 1
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
LATCH:SOURCE?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:LATCH:SOURCE 1

:SERIALbus<x>:SPATtern[:ANALyze]:SETup:SPOint

Function Sets or queries the analysis start point of the user-defined bus signal analysis.

Syntax :SERIALbus<x>:SPATtern[:ANALyze]:
SETup:SPOint {<NRf>}
:SERIALbus<x>:SPATtern[:ANALyze]:
SETup:SPOint?
<x> = 1 to 4
<NRf> = -5 to 5 (div), 10 div/display record length
steps

Example :SERIALBUS1:SPATTERN:ANALYZE:SETUP:
SPOINT -5
:SERIALBUS1:SPATTERN:ANALYZE:SETUP:
SPOINT?
-> :SERIALBUS1:SPATTERN:ANALYZE:
SETUP:SPOINT -5.0000000E+00

:SERIALbus<x>:SPATtern:SEARCh?

Function Queries all user-defined bus signal search settings.

Syntax :SERIALbus<x>:SPATtern:SEARCh?
<x> = 1 to 4

:SERIALbus<x>:SPATtern:SEARCh:ABORt

Function Aborts the user-defined bus signal search.

Syntax :SERIALbus<x>:SPATtern:SEARCh:ABORt
<x> = 1 to 4

Example :SERIALBUS1:SPATTERN:SEARCH:ABORT

:SERIALbus<x>:SPATtern:SEARCh:EXECute

Function Executes a user-defined bus signal search.

Syntax :SERIALbus<x>:SPATtern:SEARCh:
EXECute
<x> = 1 to 4

Example :SERIALBUS1:SPATTERN:SEARCH:EXECUTE

:SERIALbus<x>:SPATtern:SEARCh:SELe

Function Sets which detected point to display in the user-defined bus signal search zoom window and queries the zoom position of the detected point.

Syntax :SERIALbus<x>:SPATtern:SEARCh:
SELe ct {<NRf>|MAXimum}
:SERIALbus<x>:SPATtern:SEARCh:
SELe ct?
<x> = 1 to 4
<NRf> = 0 to 49999

Example :SERIALBUS1:SPATTERN:SEARCH:SELECT 1
:SERIALBUS1:SPATTERN:SEARCH:SELECT?
-> :SERIALBUS1:SPATTERN:SEARCH:
SELECT 1.50000000

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:SPATtern:SEARCh:SELe ct? MAXimum

Function Queries the number of detected points in the user-defined bus signal search.

Syntax :SERIALbus<x>:SPATtern:SEARCh:
SELe ct? {MAXimum}
<x> = 1 to 4

Example :SERIALBUS1:SPATTERN:SEARCH:
SELECT? MAXIMUM
-> :SERIALBUS1:SPATTERN:SEARCH:
SELECT 100

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:SPATtern:SEARCh:SETup?

Function Queries all User-defined bus signal search setup settings.

Syntax :SERIALbus<x>:SPATtern:SEARCh:SETup?
<x> = 1 to 4

:SERIALbus<x>:SPATtern:SEARCh:SETup:BI TSize

Function Sets or queries the user-defined bus signal search bit length.

Syntax :SERIALbus<x>:SPATtern:SEARCh:SETup:
BITSize {<NRf>}
:SERIALbus<x>:SPATtern:SEARCh:SETup:
BITSize?
<x> = 1 to 4
<NRf> = 1 to 128

Example :SERIALBUS1:SPATTERN:SEARCH:SETUP:
BITSIZE 1
:SERIALBUS1:SPATTERN:SEARCH:SETUP:
BITSIZE?
-> :SERIALBUS1:SPATTERN:SEARCH:
SETUP:BITSIZE 1

:SERIALbus<x>:SPATtern:SEARch:SETup:HEXa

Function Sets the data conditions of the user-defined bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:SPATtern:SEARch:SETup:HEXa {<String>}
<x> = 1 to 4

Example :SERIALBUS1:SPATtern:SEARch:SETup:HEXa "12"

:SERIALbus<x>:SPATtern:SEARch:SETup:PATtern

Function Sets or queries the data conditions of the user-defined bus signal searching in binary notation.

Syntax :SERIALbus<x>:SPATtern:SEARch:SETup:PATtern {<String>}
:SERIALbus<x>:SPATtern:SEARch:SETup:PATtern?
<x> = 1 to 4

Example :SERIALBUS1:SPATtern:SEARch:SETup:PATtern "00110101"
:SERIALBUS1:SPATtern:SEARch:SETup:PATtern?
-> :SERIALBUS1:SPATtern:SEARch:SETup:PATtern "00110101"

:SERIALbus<x>:SPI?

Function Queries all SPI bus signal analysis and search settings.

Syntax :SERIALbus<x>:SPI?
<x> = 1 to 4

:SERIALbus<x>:SPI:ANALyze?

Function Queries all SPI bus signal analysis settings.

Syntax :SERIALbus<x>:SPI:ANALyze?
<x> = 1 to 4

:SERIALbus<x>:SPI[:ANALyze]:SETup?

Function Queries all SPI bus signal analysis bus settings.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup?
<x> = 1 to 4

:SERIALbus<x>:SPI[:ANALyze]:SETup:BITorder

Function Sets or queries the bit order of the SPI bus signal analysis data.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:BITorder {MSBFirst|LSBFirst}
:SERIALbus<x>:SPI[:ANALyze]:SETup:BITorder?
<x> = 1 to 4

Example :SERIALBUS1:SPI:ANALyze:SETup:BITorder MSBFIRST
:SERIALBUS1:SPI:ANALyze:SETup:BITorder?
-> :SERIALBUS1:SPI:ANALyze:SETup:BITorder MSBFIRST

:SERIALbus<x>:SPI[:ANALyze]:SETup:CLOCK?

Function Queries all SPI bus signal analysis clock signal settings.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:CLOCK?
<x> = 1 to 4

:SERIALbus<x>:SPI[:ANALyze]:SETup:CLOCK:POLarity

Function Sets or queries the slope of the SPI bus signal analysis clock signal.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:CLOCK:POLarity {FALL|RISE}
:SERIALbus<x>:SPI[:ANALyze]:SETup:CLOCK:POLarity?
<x> = 1 to 4

Example :SERIALBUS1:SPI:ANALyze:SETup:CLOCK:POLarity FALL
:SERIALBUS1:SPI:ANALyze:SETup:CLOCK:POLarity?
-> :SERIALBUS1:SPI:ANALyze:SETup:CLOCK:POLarity FALL

5.24 SERIALbus group

:SERIALbus<x>:SPI[:ANALyze]:SETup:CLOCK:SOURCE

Function Sets or queries the SPI bus signal analysis clock signal.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:
CLOCK:SOURCE {<NRf>|BIT<y>|MATH<y>}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
CLOCK:SOURCE?
<x> = 1 to 4
<NRf> = 1 to 4
BIT<y>'s <y> = 1 to 8
MATH<y>'s <y> = 1 to 4

Example :SERIALBUS1:SPI:ANALYZE:SETUP:CLOCK:
SOURCE 1
:SERIALBUS1:SPI:ANALYZE:SETUP:CLOCK:
SOURCE?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:
CLOCK:SOURCE 1

:SERIALbus<x>:SPI[:ANALyze]:SETup:CS?

Function Queries all SPI bus signal analysis chip select signal settings.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:
CS?
<x> = 1 to 4

:SERIALbus<x>:SPI[:ANALyze]:SETup:CS:ACTIVE

Function Sets or queries the active state of the SPI bus signal analysis chip select signal.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:
CS:ACTIVE {HIGH|LOW}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
CS:ACTIVE?
<x> = 1 to 4

Example :SERIALBUS1:SPI:ANALYZE:SETUP:CS:
ACTIVE HIGH
:SERIALBUS1:SPI:ANALYZE:SETUP:CS:
ACTIVE?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:CS:
ACTIVE HIGH

:SERIALbus<x>:SPI[:ANALyze]:SETup:CS:SOURCE

Function Sets or queries the SPI bus signal analysis chip select signal.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:
CS:SOURCE {<NRf>|BIT<y>|MATH<y>}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
CS:SOURCE?
<x> = 1 to 4
<NRf> = 1 to 4
BIT<y>'s <y> = 1 to 8
MATH<y>'s <y> = 1 to 4

Example :SERIALBUS1:SPI:ANALYZE:SETUP:CS:
SOURCE 1
:SERIALBUS1:SPI:ANALYZE:SETUP:CS:
SOURCE?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:CS:
SOURCE 1

:SERIALbus<x>:SPI[:ANALyze]:SETup:DATA<y>?

Function Queries all SPI bus signal analysis data signal settings.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:
DATA<y>?
<x> = 1 to 4
<y> = 1 to 2

:SERIALbus<x>:SPI[:ANALyze]:SETup:DATA<y>:SOURCE

Function Sets or queries the SPI bus signal analysis data signal.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:
DATA<y>:SOURCE {<NRf>|BIT<z>|
MATH<z>}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
DATA<y>:SOURCE?
<x> = 1 to 4
<y> = 1 to 2
<NRf> = 1 to 4
BIT<z>'s <z> = 1 to 8
MATH<z>'s <z> = 1 to 4

Example :SERIALBUS1:SPI:ANALYZE:SETUP:DATA1:
SOURCE 1
:SERIALBUS1:SPI:ANALYZE:SETUP:DATA1:
SOURCE?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:
DATA1:SOURCE 1

:SERIALbus<x>:SPI[:ANALyze]:SETup:FIELD

Function Sets or queries the field size of the SPI bus signal analysis data.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:
FIELD {<NRF>}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
FIELD?
<x> = 1 to 4
<NRF> = 4 to 32

Example :SERIALBUS1:SPI:ANALYZE:SETUP:
FIELD 4
:SERIALBUS1:SPI:ANALYZE:SETUP:FIELD?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:
FIELD 4

:SERIALbus<x>:SPI[:ANALyze]:SETup:GROUPing

Function Sets or queries the grouping on/off state for when the SPI bus signal analysis chip select signal is not available.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:
GROUPing {<Boolean>}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
GROUPing?
<x> = 1 to 4

Example :SERIALBUS1:SPI:ANALYZE:SETUP:
GROUPING ON
:SERIALBUS1:SPI:ANALYZE:SETUP:
GROUPING?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:
GROUPING 1

:SERIALbus<x>:SPI[:ANALyze]:SETup:ITIMe

Function Sets or queries the idle time for when the SPI bus signal analysis chip select signal is not available.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:
ITIME {<Time>}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
ITIME?
<x> = 1 to 4
<Time> = 10 ns to 1 ms

Example :SERIALBUS1:SPI:ANALYZE:SETUP:
ITIME 2US
:SERIALBUS1:SPI:ANALYZE:SETUP:ITIME?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:
ITIME 2.000000E-06

:SERIALbus<x>:SPI[:ANALyze]:SETup:MODE

Function Sets or queries the wiring system (three-wire system or four-wire system) of the SPI bus signal analysis data signal.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:
MODE {WIRE3|WIRE4}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
MODE?
<x> = 1 to 4

Example :SERIALBUS1:SPI:ANALYZE:SETUP:
MODE WIRE3
:SERIALBUS1:SPI:ANALYZE:SETUP:MODE?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:
MODE WIRE3

:SERIALbus<x>:SPI[:ANALyze]:SETup:MSBLsb

Function Sets or queries the MSB/LSB bit for SPI bus signal analysis data.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:
MSBLsb {<NRF>,<NRF>}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
MSBLsb?
<x> = 1 to 4
<NRF> = 0 to 31

Example :SERIALBUS1:SPI:ANALYZE:SETUP:
MSBLSB 7,0
:SERIALBUS1:SPI:ANALYZE:SETUP:
MSBLSB?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:
MSBLSB 7,0

:SERIALbus<x>:SPI[:ANALyze]:SETup:TYPE

Function Sets or queries the SPI bus signal analysis data type.

Syntax :SERIALbus<x>:SPI[:ANALyze]:SETup:
TYPE {BYTE|USER}
:SERIALbus<x>:SPI[:ANALyze]:SETup:
TYPE?
<x> = 1 to 4

Example :SERIALBUS1:SPI:ANALYZE:SETUP:
TYPE BYTE
:SERIALBUS1:SPI:ANALYZE:SETUP:TYPE?
-> :SERIALBUS1:SPI:ANALYZE:SETUP:
TYPE BYTE

:SERIALbus<x>:SPI:DETail?

Function Queries all SPI bus signal analysis result list settings.

Syntax :SERIALbus<x>:SPI:DETail?
<x> = 1 to 4

5.24 SERIALbus group

:SERIALbus<x>:SPI:DETAil:DISPlay

Function Sets or queries the display mode for the SPI bus signal analysis result list.

Syntax :SERIALbus<x>:SPI:DETAil:
DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:SPI:DETAil:DISPlay?
<x> = 1 to 4

Example :SERIALBUS1:SPI:DETAIL:DISPLAY FULL
:SERIALBUS1:SPI:DETAIL:DISPLAY?
-> :SERIALBUS1:SPI:DETAIL:
DISPLAY FULL

:SERIALbus<x>:SPI:DETAil:LIST:ITEM?

Function Queries all items that will be displayed in the SPI bus signal analysis result list.

Syntax :SERIALbus<x>:SPI:DETAil:LIST:ITEM?
<x> = 1 to 4

Example :SERIALBUS1:SPI:DETAIL:LIST:ITEM?
-> :SERIALBUS1:SPI:DETAIL:LIST:
ITEM "No.,Time(ms),Data,,"

:SERIALbus<x>:SPI:DETAil:LIST:VALue?

Function Queries all the data for the specified analysis number in the SPI bus signal analysis result list.

Syntax :SERIALbus<x>:SPI:DETAil:LIST:
VALue? {<NRf>|MAXimum|MINimum}
<x> = 1 to 4
<NRf> = See the Features Guide for this information.

Example :SERIALBUS1:SPI:DETAIL:LIST:VALUE? 0
-> :SERIALBUS1:SPI:DETAIL:LIST:
VALUE "0,0.077002,Data1,
00 00 00 00 00 00 00 00,"

:SERIALbus<x>:SPI:SEARCh?

Function Queries all SPI bus signal search settings.

Syntax :SERIALbus<x>:SPI:SEARCh?
<x> = 1 to 4

:SERIALbus<x>:SPI:SEARCh:ABORT

Function Aborts the SPI bus signal search.

Syntax :SERIALbus<x>:SPI:SEARCh:ABORT
<x> = 1 to 4

Example :SERIALBUS1:SPI:SEARCH:ABORT

:SERIALbus<x>:SPI:SEARCh:EXECute

Function Executes an SPI bus signal search.

Syntax :SERIALbus<x>:SPI:SEARCh:EXECute
<x> = 1 to 4

Example :SERIALBUS1:SPI:SEARCH:EXECUTE

:SERIALbus<x>:SPI:SEARCh:SELEct

Function Sets which detected point to display in the SPI bus signal search zoom window and queries the zoom position of the detected point.

Syntax :SERIALbus<x>:SPI:SEARCh:
SELEct {<NRf>|MAXimum}
:SERIALbus<x>:SPI:SEARCh:SELEct?
<x> = 1 to 4
<NRf> = 0 to 49999

Example :SERIALBUS1:SPI:SEARCH:SELECT 1
:SERIALBUS1:SPI:SEARCH:SELECT?
-> :SERIALBUS1:SPI:SEARCH:
SELECT 1.50000000

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:SPI:SEARCh:SELEct? MAXimum

Function Queries the number of detected points in the SPI bus signal search.

Syntax :SERIALbus<x>:SPI:SEARCh:
SELEct? {MAXimum}
<x> = 1 to 4

Example :SERIALBUS1:SPI:SEARCH:
SELECT? MAXIMUM
-> :SERIALBUS1:SPI:SEARCH:SELECT 100

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:SPI:SEARCh:SETup?

Function Queries all SPI bus signal search condition settings.

Syntax :SERIALbus<x>:SPI:SEARCh:SETup?
<x> = 1 to 4

:SERIALbus<x>:SPI:SEARCh[:SETup]:DATA<y>?

Function Queries all SPI bus signal search data settings.

Syntax :SERIALbus<x>:SPI:SEARCh[:SETup]:
DATA<y>?
<x> = 1 to 4
<y> = 1 to 2

:SERIALbus<x>:SPI:SEARCH[:SETUP]:DATA<y>:BCOUNT

Function Sets or queries the pattern comparison start position of the specified data for SPI bus signal searching.

Syntax :SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:BCOUNT {<NRf>}
:SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:BCOUNT?
<x> = 1 to 4
<y> = 1 to 2
<NRf> = 0 to 9999

Example :SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
BCOUNT 0
:SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
BCOUNT?
-> :SERIALBUS1:SPI:SEARCH:SETUP:
DATA1:BCOUNT 0

:SERIALbus<x>:SPI:SEARCH[:SETUP]:DATA<y>:CONDITION

Function Sets or queries the judgment method (true/false) of the specified data for SPI bus signal searching.

Syntax :SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:CONDITION {FALSE|TRUE}
:SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:CONDITION?
<x> = 1 to 4
<y> = 1 to 2

Example :SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
CONDITION DONTCARE
:SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
CONDITION?
-> :SERIALBUS1:SPI:SEARCH:SETUP:
DATA1:CONDITION DONTCARE

:SERIALbus<x>:SPI:SEARCH[:SETUP]:DATA<y>:DBYTE

Function Sets or queries the data size (number of bytes) of the specified data for SPI bus signal searching.

Syntax :SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:DBYTE {<NRf>}
:SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:DBYTE?
<x> = 1 to 4
<y> = 1 to 2
<NRf> = 1 to 4

Example :SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
DBYTE 1
:SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
DBYTE?
-> :SERIALBUS1:SPI:SEARCH:SETUP:
DATA1:DBYTE 1

:SERIALbus<x>:SPI:SEARCH[:SETUP]:DATA<y>:DPATTERN?

Function Queries all SPI bus signal search data settings.

Syntax :SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:DPATTERN?
<x> = 1 to 4

:SERIALbus<x>:SPI:SEARCH[:SETUP]:DATA<y>:DPATTERN:HEXA

Function Sets the specified data for SPI bus signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:DPATTERN:HEXA {<String>}
<x> = 1 to 4

Example :SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
DPATTERN:HEXA "AB"

Description The number of bytes that can be set varies depending on the :SERIALbus<x>:SPI:SEARCH[:SETUP]:DATA<y>:DBYTE setting.

:SERIALbus<x>:SPI:SEARCH[:SETUP]:DATA<y>:DPATTERN:PATTERN

Function Sets or queries the specified data for SPI bus signal searching in binary notation.

Syntax :SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:DPATTERN:PATTERN {<String>}
:SERIALbus<x>:SPI:SEARCH[:SETUP]:
DATA<y>:DPATTERN:PATTERN?
<x> = 1 to 4

Example :SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
DPATTERN:PATTERN "11001010"
:SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
DPATTERN:PATTERN?
-> :SERIALBUS1:SPI:SEARCH:SETUP:
DATA1:DPATTERN:PATTERN "11001010"

Description The number of bytes that can be set varies depending on the :SERIALbus<x>:SPI:SEARCH[:SETUP]:DATA<y>:DBYTE setting.

5.24 SERIALbus group

:SERIALbus<x>:SPI:SEARCh[:SETup]:DATA<y>:MODE

Function Sets or queries the data condition enable/disable state for SPI bus signal searching.

Syntax :SERIALbus<x>:SPI:SEARCh[:SETup]:
DATA<y>:MODE {<Boolean>}
:SERIALbus<x>:SPI:SEARCh[:SETup]:
DATA<y>:MODE?
<x> = 1 to 4
<y> = 1 to 2

Example :SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
MODE ON
:SERIALBUS1:SPI:SEARCH:SETUP:DATA1:
MODE?
-> :SERIALBUS1:SPI:SEARCH:SETUP:
DATA1:MODE 1

:SERIALbus<x>:TYPE

Function Sets or queries the search type.

Syntax :SERIALbus<x>:TYPE {CANBus|CANFDbus|
CXPIbus|FLEXray|I2Cbus|LINbus|PSI5|
SENT|SPATtern|SPIbus|UART}
:SERIALbus<x>:TYPE?
<x> = 1 to 4

Example :SERIALBUS1:TYPE CANBUS
:SERIALBUS1:TYPE?
-> :SERIALBUS1:TYPE CANBUS

:SERIALbus<x>:UART?

Function Queries all UART signal analysis and search settings.

Syntax :SERIALbus<x>:UART?
<x> = 1 to 4

:SERIALbus<x>:UART:ANALyze?

Function Queries all UART signal analysis settings.

Syntax :SERIALbus<x>:UART:ANALyze?
<x> = 1 to 4

:SERIALbus<x>:UART[:ANALyze]:SETup?

Function Queries all bus setup settings for UART signal analysis.

Syntax :SERIALbus<x>:UART[:ANALyze]:SETup?
<x> = 1 to 4

:SERIALbus<x>:UART[:ANALyze]:SETup:BIT order

Function Sets or queries the UART signal analysis bit order.

Syntax :SERIALbus<x>:UART[:ANALyze]:SETup:
BITOrder {MSBFirst|LSBFirst}
:SERIALbus<x>:UART[:ANALyze]:SETup:
BITOrder?
<x> = 1 to 4

Example :SERIALBUS1:UART:ANALYZE:SETUP:
BITORDER MSBFIRST
:SERIALBUS1:UART:ANALYZE:SETUP:
BITORDER?
-> :SERIALBUS1:UART:ANALYZE:SETUP:
BITORDER MSBFIRST

:SERIALbus<x>:UART[:ANALyze]:SETup:BRATe

Function Sets or queries the UART signal analysis bit rate (data transfer rate).

Syntax :SERIALbus<x>:UART[:ANALyze]:SETup:
BRATe {<Nrf>|USER,<Nrf>}
:SERIALbus<x>:UART[:ANALyze]:SETup:
BRATe?
<x> = 1 to 4
<Nrf> = 1200, 2400, 4800, 9600, 19200
USER <Nrf> = See the Features Guide for this information.

Example :SERIALBUS1:UART:ANALYZE:SETUP:
BRATE 2400
:SERIALBUS1:UART:ANALYZE:SETUP:
BRATE?
-> :SERIALBUS1:UART:ANALYZE:SETUP:
BRATE 2400

:SERIALbus<x>:UART[:ANALyze]:SETup:BSpace

Function Sets or queries the UART signal analysis byte space.

Syntax :SERIALbus<x>:UART[:ANALyze]:SETup:
BSpace {<Time>|Minimum}
:SERIALbus<x>:UART[:ANALyze]:SETup:
BSpace?
<x> = 1 to 4
<Time> = See the Features Guide for this information.

Example :SERIALBUS1:UART:ANALYZE:SETUP:
BSPACE 0.1S
:SERIALBUS1:UART:ANALYZE:SETUP:
BSPACE?
-> :SERIALBUS1:UART:ANALYZE:SETUP:
BSPACE 1.000E-01

:SERIALbus<x>:UART[:ANALyze]:SETup:GROUPing

Function Sets or queries on/off state of UART signal analysis grouping.

Syntax :SERIALbus<x>:UART[:ANALyze]:SETup:GROUPing {<Boolean>}
:SERIALbus<x>:UART[:ANALyze]:SETup:GROUPing?
<x> = 1 to 4

Example :SERIALBUS1:UART:ANALYZE:SETUP:GROUPING ON
:SERIALBUS1:UART:ANALYZE:SETUP:GROUPING?
-> :SERIALBUS1:UART:ANALYZE:SETUP:GROUPING 1

:SERIALbus<x>:UART[:ANALyze]:SETup:POLarity

Function Sets or queries the UART signal analysis polarity.

Syntax :SERIALbus<x>:UART[:ANALyze]:SETup:POLarity {NEGAtive|POSitive}
:SERIALbus<x>:UART[:ANALyze]:SETup:POLarity?
<x> = 1 to 4

Example :SERIALBUS1:UART:ANALYZE:SETUP:POLARITY NEGATIVE
:SERIALBUS1:UART:ANALYZE:SETUP:POLARITY?
-> :SERIALBUS1:UART:ANALYZE:SETUP:POLARITY NEGATIVE

:SERIALbus<x>:UART[:ANALyze]:SETup:SOURce

Function Sets or queries the UART signal analysis signal.

Syntax :SERIALbus<x>:UART[:ANALyze]:SETup:SOURce {<NRf>|BIT<y>|MATH<y>}
:SERIALbus<x>:UART[:ANALyze]:SETup:SOURce?
<x> = 1 to 4
<NRf> = 1 to 4
BIT<y>'s <y> = 1 to 8
MATH<y>'s <y> = 1 to 4

Example :SERIALBUS1:UART:ANALYZE:SETUP:SOURCE 1
:SERIALBUS1:UART:ANALYZE:SETUP:SOURCE?
-> :SERIALBUS1:UART:ANALYZE:SETUP:SOURCE 1

:SERIALbus<x>:UART[:ANALyze]:SETup:SPOint

Function Sets or queries the UART signal analysis sample point.

Syntax :SERIALbus<x>:UART[:ANALyze]:SETup:SPOint {<NRf>}
:SERIALbus<x>:UART[:ANALyze]:SETup:SPOint?
<x> = 1 to 4
<NRf> = 18.8 to 90.6

Example :SERIALBUS1:UART:ANALYZE:SETUP:SPOINT 18.8
:SERIALBUS1:UART:ANALYZE:SETUP:SPOINT?
-> :SERIALBUS1:UART:ANALYZE:SETUP:SPOINT 18.8

:SERIALbus<x>:UART:DETAil?

Function Queries all UART signal analysis result list settings.

Syntax :SERIALbus<x>:UART:DETAil?
<x> = 1 to 4

:SERIALbus<x>:UART:DETAil:DISPlay

Function Sets or queries the display mode for the UART signal analysis result list.

Syntax :SERIALbus<x>:UART:DETAil:DISPlay {FULL|LOWer|UPPer}
:SERIALbus<x>:UART:DETAil:DISPlay?
<x> = 1 to 4

Example :SERIALBUS1:UART:DETAIL:DISPLAY FULL
:SERIALBUS1:UART:DETAIL:DISPLAY?
-> :SERIALBUS1:UART:DETAIL:DISPLAY FULL

:SERIALbus<x>:UART:DETAil:LIST:ITEM?

Function Queries all items that will be displayed in the UART signal analysis result list.

Syntax :SERIALbus<x>:UART:DETAil:LIST:ITEM?
<x> = 1 to 4

Example :SERIALBUS1:UART:DETAIL:LIST:ITEM?
-> :SERIALBUS1:UART:DETAIL:LIST:ITEM "No.,Time(ms),Data(HEX),Data(ASCII),Information"

5.24 SERIALbus group

:SERIALbus<x>:UART:DETAil:LIST:VALue?

Function Queries all the data for the specified analysis number in the UART signal analysis result list.

Syntax :SERIALbus<x>:UART:DETAil:LIST:
VALue? {<NRf>|MAXimum|MINimum}
<x> = 1 to 4

<NRf> = See the Features Guide for this information.

Example :SERIALBUS1:UART:DETAIL:LIST:
VALUE? 0
-> :SERIALBUS1:UART:DETAIL:LIST:
VALUE "0,-0.5720,3A 74 69 D2,":ti.",,"

:SERIALbus<x>:UART:SEARCh?

Function Queries all UART signal search settings.

Syntax :SERIALbus<x>:UART:SEARCh?
<x> = 1 to 4

:SERIALbus<x>:UART:SEARCh:ABORT

Function Aborts the UART signal search.

Syntax :SERIALbus<x>:UART:SEARCh:ABORT
<x> = 1 to 4

Example :SERIALBUS1:UART:SEARCH:ABORT

:SERIALbus<x>:UART:SEARCh:EXECute

Function Executes a UART signal search.

Syntax :SERIALbus<x>:UART:SEARCh:EXECute
<x> = 1 to 4

Example :SERIALBUS1:UART:SEARCH:EXECUTE

:SERIALbus<x>:UART:SEARCh:SELEct

Function Sets which detected point to display in the UART signal search zoom window and queries the zoom position of the detected point.

Syntax :SERIALbus<x>:UART:SEARCh:
SELEct {<NRf>|MAXimum}
:SERIALbus<x>:UART:SEARCh:SELEct?
<x> = 1 to 4
<NRf> = 0 to 49999

Example :SERIALBUS1:UART:SEARCH:SELECT 1
:SERIALBUS1:UART:SEARCH:SELECT?
-> :SERIALBUS1:UART:SEARCH:
SELECT 1.50000000

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:UART:SEARCh:SELEct? Maximum

Function Queries the number of detected points in the UART signal search.

Syntax :SERIALbus<x>:UART:SEARCh:
SELEct? {MAXimum}
<x> = 1 to 4

Example :SERIALBUS1:UART:SEARCH:
SELECT? MAXIMUM
-> :SERIALBUS1:UART:SEARCH:
SELECT 100

Description If there are no detected points, the instrument returns "NAN."

:SERIALbus<x>:UART:SEARCh:SETUp?

Function Queries all search conditions for UART signal searching.

Syntax :SERIALbus<x>:UART:SEARCh:SETUp?
<x> = 1 to 4

:SERIALbus<x>:UART:SEARCh:SETUp:DATA?

Function Queries all UART signal search data settings.

Syntax :SERIALbus<x>:UART:SEARCh:SETUp:
DATA?
<x> = 1 to 4

:SERIALbus<x>:UART:SEARCh:SETUp:DATA:AScii

Function Sets the UART signal search data in ASCII format.

Syntax :SERIALbus<x>:UART:SEARCh:SETUp:
DATA:AScii {<String>}
<x> = 1 to 4

Example :SERIALBUS1:UART:SEARCH:SETUP:DATA:
ASCII "TEST"

:SERIALbus<x>:UART:SEARCh:SETUp:DATA:CONDition

Function Sets or queries the data judgment method (true/false) for UART signal searching.

Syntax :SERIALbus<x>:UART:SEARCh:SETUp:
DATA:CONDition {DONTcare|TRUE}
:SERIALbus<x>:UART:SEARCh:SETUp:
DATA:CONDition?
<x> = 1 to 4

Example :SERIALBUS1:UART:SEARCH:SETUP:DATA:
CONDITION DONTCARE
:SERIALBUS1:UART:SEARCH:SETUP:DATA:
CONDITION?
-> :SERIALBUS1:UART:SEARCH:SETUP:
DATA:CONDITION DONTCARE

:SERIALbus<x>:UART:SEARCH:SETUP:DATA:CSensitive

Function Sets or queries whether to distinguish uppercase and lowercase letters in ASCII data for the UART signal search.

Syntax :SERIALbus<x>:UART:SEARCH:SETUP:
DATA:CSensitive {<Boolean>}
:SERIALbus<x>:UART:SEARCH:SETUP:
DATA:CSensitive?

Example :SERIALBUS1:UART:SEARCH:SETUP:DATA:
CSensitive ON
:SERIALBUS1:UART:SEARCH:SETUP:DATA:
CSensitive?
-> :SERIALBUS1:UART:SEARCH:SETUP:
DATA:CSensitive 1

:SERIALbus<x>:UART:SEARCH:SETUP:DATA:DBYTE

Function Sets or queries the number of data bytes for UART signal searching.

Syntax :SERIALbus<x>:UART:SEARCH:SETUP:
DATA:DBYTE {<Nrf>}
:SERIALbus<x>:UART:SEARCH:SETUP:
DATA:DBYTE?

Example :SERIALBUS1:UART:SEARCH:SETUP:DATA:
DBYTE 1
:SERIALBUS1:UART:SEARCH:SETUP:DATA:
DBYTE?
-> :SERIALBUS1:UART:SEARCH:SETUP:
DATA:DBYTE 1

:SERIALbus<x>:UART:SEARCH:SETUP:DATA:DPATTERN?

Function Queries all UART signal search data settings.

Syntax :SERIALbus<x>:UART:SEARCH:SETUP:
DATA:DPATTERN?
<x> = 1 to 4

:SERIALbus<x>:UART:SEARCH:SETUP:DATA:DPATTERN:HEXA

Function Sets the data for UART signal searching in hexadecimal notation.

Syntax :SERIALbus<x>:UART:SEARCH:SETUP:
DATA:DPATTERN:HEXA {<String>}
<x> = 1 to 4

Example :SERIALBUS1:UART:SEARCH:SETUP:DATA:
DPATTERN:HEXA "12"

:SERIALbus<x>:UART:SEARCH:SETUP:DATA:DPATTERN:PATTERN

Function Sets or queries the data for UART signal searching in binary notation.

Syntax :SERIALbus<x>:UART:SEARCH:SETUP:
DATA:DPATTERN:PATTERN {<String>}
:SERIALbus<x>:UART:SEARCH:SETUP:
DATA:DPATTERN:PATTERN?
<x> = 1 to 4

Example :SERIALBUS1:UART:SEARCH:SETUP:DATA:
DPATTERN:PATTERN "00110101"
:SERIALBUS1:UART:SEARCH:SETUP:DATA:
DPATTERN:PATTERN?
-> :SERIALBUS1:UART:SEARCH:SETUP:
DATA:DPATTERN:PATTERN "00110101"

:SERIALbus<x>:UART:SEARCH:SETUP:DATA:PFORMAT

Function Sets or queries the data input format, which is one of the ID and data conditions, for UART signal searching.

Syntax :SERIALbus<x>:UART:SEARCH:SETUP:
DATA:PFORMAT {ASCII|PATTERN}
:SERIALbus<x>:UART:SEARCH:SETUP:
DATA:PFORMAT?
<x> = 1 to 4

Example :SERIALBUS1:UART:SEARCH:SETUP:DATA:
PFORMAT PATTERN
:SERIALBUS1:UART:SEARCH:SETUP:DATA:
PFORMAT?
-> :SERIALBUS1:UART:SEARCH:SETUP:
DATA:PFORMAT PATTERN

:SERIALbus<x>:UART:SEARCH:SETUP:ERROR?

Function Queries all UART signal search error settings.

Syntax :SERIALbus<x>:UART:SEARCH:SETUP:
ERROR?
<x> = 1 to 4

:SERIALbus<x>:UART:SEARCH:SETUP:ERROR:FRAMING

Function Sets or queries the UART signal search framing error setting.

Syntax :SERIALbus<x>:UART:SEARCH:SETUP:
ERROR:FRAMING {<Boolean>}
:SERIALbus<x>:UART:SEARCH:SETUP:
ERROR:FRAMING?
<x> = 1 to 4

Example :SERIALBUS1:UART:SEARCH:SETUP:ERROR:
FRAMING ON
:SERIALBUS1:UART:SEARCH:SETUP:ERROR:
FRAMING?
-> :SERIALBUS1:UART:SEARCH:SETUP:
ERROR:FRAMING 1

5.24 SERIALbus group

:SERIALbus<x>:UART:SEARCH:SETUP:ERROR:PARity

Function Sets or queries the UART signal search parity error setting.

Syntax :SERIALbus<x>:UART:SEARCH:SETUP:
ERROR:PARity {<Boolean>}
:SERIALbus<x>:UART:SEARCH:SETUP:
ERROR:PARity?
<x> = 1 to 4

Example :SERIALBUS1:UART:SEARCH:SETUP:ERROR:
PARITY ON
:SERIALBUS1:UART:SEARCH:SETUP:ERROR:
PARITY?
-> :SERIALBUS1:UART:SEARCH:SETUP:
ERROR:PARITY 1

:SERIALbus<x>:UART:SEARCH:SETUP:ERROR:PMODE

Function Sets or queries the UART signal search parity mode setting.

Syntax :SERIALbus<x>:UART:SEARCH:SETUP:
ERROR:PMODE {EVEN|ODD}
:SERIALbus<x>:UART:SEARCH:SETUP:
ERROR:PMODE?
<x> = 1 to 4

Example :SERIALBUS1:UART:SEARCH:SETUP:ERROR:
PMODE EVEN
:SERIALBUS1:UART:SEARCH:SETUP:ERROR:
PMODE?
-> :SERIALBUS1:UART:SEARCH:SETUP:
ERROR:PMODE EVEN

:SERIALbus<x>:UART:SEARCH:SETUP:FORMAT

Function Sets or queries the UART signal search format setting.

Syntax :SERIALbus<x>:UART:SEARCH:SETUP:
FORMAT {BIT7Parity|BIT8Parity|
BIT8Noparity}
:SERIALbus<x>:UART:SEARCH:SETUP:
FORMAT?
<x> = 1 to 4

Example :SERIALBUS1:UART:SEARCH:SETUP:
FORMAT BIT7PARITY
:SERIALBUS1:UART:SEARCH:SETUP:
FORMAT?
-> :SERIALBUS1:UART:SEARCH:SETUP:
FORMAT BIT7PARITY

:SERIALbus<x>:UART:SEARCH:SETUP:MODE

Function Sets or queries the UART signal search mode setting.

Syntax :SERIALbus<x>:UART:SEARCH:SETUP:
MODE {DATA|EDATA|ERROR}
:SERIALbus<x>:UART:SEARCH:SETUP:
MODE?
<x> = 1 to 4

Example :SERIALBUS1:UART:SEARCH:SETUP:
MODE DATA
:SERIALBUS1:UART:SEARCH:SETUP:MODE?
-> :SERIALBUS1:UART:SEARCH:SETUP:
MODE DATA

:SERIALbus<x>:ZLINKage

Function Sets or queries the linkage (on/off) between the result numbers and zoom positions of the serial bus signal analysis results.

Syntax :SERIALbus<x>:ZLINKage {<Boolean>}
:SERIALbus<x>:ZLINKage?
<x> = 1 to 4

Example :SERIALBUS1:ZLINKAGE ON
:SERIALBUS1:ZLINKAGE?
-> :SERIALBUS1:ZLINKAGE 1

5.25 SNAP Group

:SNAP

Function Takes a snapshot.

Syntax :SNAP

Example :SNAP

5.26 SStart Group

:SStart?

Function	Sets the trigger mode to single, starts waveform acquisition. The instrument returns 0 if the waveform acquisition is stopped within the specified timeout. Otherwise, the instrument returns 1.
Syntax	<pre>:SStart? {<NRf>}</pre> <NRf> = 1 to 36000 (timeout value in 100 ms steps; start and wait) <NRf> = 0 (start only, no timeout) <NRf> = -36000 to -1 (timeout value in 100 ms steps; wait without starting)
Example	<pre>:SStart? 10 -> :SStart 1</pre>
Description	• If the specified timeout is a positive value, the instrument starts waveform acquisition in single trigger mode and waits for the acquisition to be stopped within the specified timeout period. • If the specified timeout is 0, the instrument starts waveform acquisition and returns 0 without waiting. • If the specified timeout is a negative value, the instrument does not start waveform acquisition and simply waits for the acquisition to be stopped within the specified timeout period.

5.27 START Group

:START

Function Starts waveform acquisition.

Syntax :START

Example :START

Description Stop the waveform acquisition using “:STOP”.

5.28 STATus Group

The commands in this group are used to make settings and queries related to the communication status feature. There are no front panel keys that correspond to the commands in this group. For information about status reports, see chapter 6.

:STATus?

Function Queries all the settings of the communication status feature.

Syntax :STATus?

:STATus:CONDition?

Function Queries the contents of the condition register.

Syntax :STATus:CONDition?

Example :STATUS:CONDITION?
-> 16

Description For information about the condition register, see chapter 6, "Status Reports."

:STATus:EES

Function Sets or queries the extended event enable register.

Syntax :STATus:EES {<Register>}
:STATus:EES?

Example :STATUS:EES 257
:STATUS:EES?
-> :STATUS:EES 257

Description For information about the extended event enable register, see chapter 6, "Status Reports."

:STATus:EESR?

Function Queries the contents of the extended event register and clears the register.

Syntax :STATus:EESR?

Example :STATUS:EESR?
-> 1

Description For information about the extended event register, see chapter 6, "Status Reports."

:STATus:ERRor?

Function Queries the error code and message of the last error that has occurred (top of the error queue).

Syntax :STATus:ERRor?

Example :STATUS:ERROR?
-> 113,"Undefined header"

:STATus:FILTer<x>

Function Sets or queries the transition filter.

Syntax :STATus:FILTer<x> {BOTH|FALL|NEVer|RISE}
:STATus:FILTer<x>?
<x> = 1 to 16

Example :STATUS:FILTER2 RISE
:STATUS:FILTER2?
-> :STATUS:FILTER2 RISE

Description For information about the transition filter, see chapter 6, "Status Reports."

:STATus:QENable

Function Sets or queries whether messages other than errors will be stored to the error queue.

Syntax :STATus:QENable {<Boolean>}
:STATus:QENable?

Example :STATUS:QENABLE ON
:STATUS:QENABLE?
-> :STATUS:QENABLE 1

:STATus:QMESsage

Function Sets or queries whether message information will be attached to the response to the :STATus:ERRor? query.

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

5.29 STOP Group

:STOP

Function Stops waveform acquisition.

Syntax :STOP

Example :STOP

Description Start the waveform acquisition using “:START”.

5.30 STORe Group

:STORe?

Function Queries all settings related to setup data in the internal memory.

Syntax :STORe?

:STORe:SETup<x>?

Function Queries all settings related to the setup data in the specified location of the internal memory.

Syntax :STORe:SETup<x>?
<x> = 1 to 3

:STORe:SETup<x>:COMMeNt

Function Sets or queries the comment for the setup data that is stored to the specified location in the internal memory.

Syntax :STORe:SETup<x>:COMMeNt {<String>}
:STORe:SETup<x>:COMMeNt?
<x> = 1 to 3
<String> = Up to 16 characters

Example :STORe:SETUP1:COMMeNt "WAVE1"
:STORe:SETUP1:COMMeNt?
-> :STORe:SETUP1:COMMeNt "WAVE1"

:STORe:SETup<x>:DATE?

Function Queries the date and time of the setup data that is stored to the specified location in the internal memory.

Syntax :STORe:SETup<x>:DATE?
<x> = 1 to 3

Example :STORe:SETUP1:DATE?
-> :STORe:SETUP1:
DATE "2018/11/11 10:56:22"

:STORe:SETup<x>:EXECute

Function Saves setup data to the specified location in the internal memory.

Syntax :STORe:SETup<x>:EXECute
<x> = 1 to 3

Example :STORe:SETUP1:EXECUTE

:STORe:SETup<x>:LOCK

Function Sets or queries the protection on/off state for the setup data stored to the specified location in the internal memory.

Syntax :STORe:SETup<x>:LOCK {<Boolean>}
:STORe:SETup<x>:LOCK?
<x> = 1 to 3

Example :STORe:SETUP1:LOCK ON
:STORe:SETUP1:LOCK?
-> :STORe:SETUP1:LOCK 1

5.31 SYSTem Group

:SYSTem?

Function Queries all system settings.
Syntax :SYSTem?

:SYSTem:BEEP

Function Generates a beep sound.
Syntax :SYSTem:BEEP
Example :SYSTEM:BEEP

:SYSTem:CLICk

Function Sets or queries whether click sounds are produced.
Syntax :SYSTem:CLICk {<Boolean>}
:SYSTem:CLICk?
Example :SYSTEM:CLICk ON
:SYSTEM:CLICk?
-> :SYSTEM:CLICk 1

:SYSTem:CLOCK?

Function Queries all date/time settings.
Syntax :SYSTem:CLOCK?

:SYSTem:CLOCK:DATE

Function Sets or queries the date.
Syntax :SYSTem:CLOCK:DATE {<String>}
:SYSTem:CLOCK:DATE?
<String> = YYYY/MM/DD. See the Features Guide for this information.
Example :SYSTEM:CLOCK:DATE "2024/04/09"
:SYSTEM:CLOCK:DATE?
-> :SYSTEM:CLOCK:DATE "2024/04/09"

:SYSTem:CLOCK:FORMat

Function Sets or queries the date format.
Syntax :SYSTem:CLOCK:FORMat {<NRf>}
:SYSTem:CLOCK:FORMat?
<NRf> = 1 to 4
Example :SYSTEM:CLOCK:FORMat 1
:SYSTEM:CLOCK:FORMat?
-> :SYSTEM:CLOCK:FORMat 1

:SYSTem:CLOCK:MODE

Function Sets or queries whether the date and time are displayed.
Syntax :SYSTem:CLOCK:MODE {<Boolean>}
:SYSTem:CLOCK:MODE?
Example :SYSTEM:CLOCK:MODE ON
:SYSTEM:CLOCK:MODE?
-> :SYSTEM:CLOCK:MODE 1

:SYSTem:CLOCK:SNTP?

Function Queries the SNTP clock setting.
Syntax :SYSTem:CLOCK:SNTP?

:SYSTem:CLOCK:SNTP:EXECute

Function Sets the clock using SNTP.
Syntax :SYSTem:CLOCK:SNTP:EXECute
Example :SYSTEM:CLOCK:SNTP:EXECUTE

:SYSTem:CLOCK:SNTP:GMTTime

Function Sets or queries the time difference from Greenwich Mean Time.
Syntax :SYSTem:CLOCK:SNTP:GMTTime {<String>}
:SYSTem:CLOCK:SNTP:GMTTime?
<String> = HH:MM. See the Features Guide for this information.
Example :SYSTEM:CLOCK:SNTP:GMTTIME "09:00"
:SYSTEM:CLOCK:SNTP:GMTTIME?
-> :SYSTEM:CLOCK:SNTP:GMTTIME "09:00"

:SYSTem:CLOCK:TIME

Function Sets or queries the time.
Syntax :SYSTem:CLOCK:TIME {<String>}
:SYSTem:CLOCK:TIME?
<String> = HH:MM:SS. See the Features Guide for this information.
Example :SYSTEM:CLOCK:TIME "14:30:00"
:SYSTEM:CLOCK:TIME?
-> :SYSTEM:CLOCK:TIME "14:30:00"

:SYSTem:DCANcel (Delay Cancel)

Function Sets or queries whether the specified delay is applied to time measurement values.
Syntax :SYSTem:DCANcel {<Boolean>}
:SYSTem:DCANcel?
Example :SYSTEM:DCANcel ON
:SYSTEM:DCANcel?
-> :SYSTEM:DCANcel 1

5.31 SYSTem Group

:SYSTem:DLMSync?

Function Queries all synchronous operation settings.
Syntax :SYSTem:DLMSync?

:SYSTem:DLMSync:ADESkew

Function Executes auto deskewing for synchronous operation.
Syntax :SYSTem:DLMSync:ADESkew
Example :SYSTEM:DLMSync:ADESKEW

:SYSTem:DLMSync:CONNect?

Function Starts synchronous operation and queries for the results.
Syntax :SYSTem:DLMSync:CONNect?
Example :SYSTEM:DLMSync:CONNECT?
-> ::SYSTEM:DLMSync:CONNECT 0
Description If synchronous operation is successful, 0 is returned. If not, 1 is returned.

:SYSTem:DLMSync:DESKew

Function Sets or queries the deskewing for synchronous operation.
Syntax :SYSTem:DLMSync:DESKew {<Time>}
:SYSTem:DLMSync:DESKew?
<Time> = 10 to 35 ns, 0.05 ns steps
Example :SYSTEM:DLMSync:DESKew 20.2E-09
:SYSTEM:DLMSync:DESKew?
-> :SYSTEM:DLMSync:DESKew 20.2000E-09

:SYSTem:DLMSync:DISConnect?

Function Ends synchronous operation and queries for the results.
Syntax :SYSTem:DLMSync:DISConnect?
Example :SYSTEM:DLMSync:DISCONNECT?
-> :SYSTEM:DLMSync:DISCONNECT 0
Description If successful, 0 is returned. If not, 1 is returned.

:SYSTem:LANGuage

Function Sets or queries the message language.
Syntax :SYSTem:LANGuage {CHINEse|ENGLish|
FRENch|GERMan|ITALian|JAPANese|
KOREan|SPANish}
:SYSTem:LANGuage?
Example :SYSTEM:LANGUAGE JAPANESE
:SYSTEM:LANGUAGE?
-> :SYSTEM:LANGUAGE JAPANESE

:SYSTem:LCD?

Function Queries all LCD settings.
Syntax :SYSTem:LCD?

:SYSTem:LCD:AUTO?

Function Queries all settings related to the LCD backlight auto-off feature.
Syntax :SYSTem:LCD:AUTO?

:SYSTem:LCD:AUTO:MODE

Function Sets or queries on/off state of the LCD backlight auto-off feature.
Syntax :SYSTem:LCD:AUTO:MODE {<Boolean>}
:SYSTem:LCD:AUTO:MODE?
Example :SYSTEM:LCD:AUTO:MODE ON
:SYSTEM:LCD:AUTO:MODE?
-> :SYSTEM:LCD:AUTO:MODE 1

:SYSTem:LCD:AUTO:TIME

Function Sets or queries the LCD backlight auto-off time.
Syntax :SYSTem:LCD:AUTO:TIME {<Nrf>}
:SYSTem:LCD:AUTO:TIME?
<Nrf> = 1 to 60 (minutes)
Example :SYSTEM:LCD:AUTO:TIME 1
:SYSTEM:LCD:AUTO:TIME?
-> :SYSTEM:LCD:AUTO:TIME 1

:SYSTem:LCD:BRIGhtness

Function Sets or queries the LCD brightness.
Syntax :SYSTem:LCD:BRIGhtness {<Nrf>}
:SYSTem:LCD:BRIGhtness?
<Nrf> = 1 to 8
Example :SYSTEM:LCD:BRIGHTNESS 2
:SYSTEM:LCD:BRIGHTNESS?
-> :SYSTEM:LCD:BRIGHTNESS 2

:SYSTem:LCD:MODE

Function Sets or queries the LCD backlight on/off state.
Syntax :SYSTem:LCD:MODE {<Boolean>}
:SYSTem:LCD:MODE?
Example :SYSTEM:LCD:MODE ON
:SYSTEM:LCD:MODE?
-> :SYSTEM:LCD:MODE 1

:SYSTem:MLANGuage

Function Sets or queries the menu language.
Syntax :SYSTem:MLANGuage {CHINEse|ENGLish|
FRENch|GERMan|ITALian|JAPANese|
KOREan|SPANish}
:SYSTem:MLANGuage?
Example :SYSTEM:MLANGUAGE JAPANESE
:SYSTEM:MLANGUAGE?
-> :SYSTEM:MLANGUAGE JAPANESE

:SYSTem:MPARameter?

Function Queries all settings related to automated measurement values of waveform parameters and cursor measurement values.

Syntax :SYSTem:MPARameter?

:SYSTem:MPARameter:FSIZE

Function Sets or queries the font size that is used to display the automatically measured waveform parameters and cursor measurement values.

Syntax :SYSTem:MPARameter:FSIZE {LARGE|SMALL}
:SYSTem:MPARameter:FSIZE?

Example :SYSTEM:MPARAMETER:FSIZE SMALL
:SYSTEM:MPARAMETER:FSIZE?
-> :SYSTEM:MPARAMETER:FSIZE SMALL

:SYSTem:MPARameter:MROWS

Function Sets or queries the number of rows of automatically measured waveform parameters and cursor measurement values to show from the menu display area.

Syntax :SYSTem:MPARameter:MROWS {<NRf>}
:SYSTem:MPARameter:MROWS?
<NRf> = 2 to 30

Example :SYSTEM:MPARAMETER:MROWS 4
:SYSTEM:MPARAMETER:MROWS?
-> :SYSTEM:MPARAMETER:MROWS 4

:SYSTem:OCANcel (Offset Cancel)

Function Sets or queries whether the specified offset voltage is applied to measurement results and computation results.

Syntax :SYSTem:OCANcel {<Boolean>}
:SYSTem:OCANcel?

Example :SYSTEM:OCANCEL ON
:SYSTEM:OCANCEL?
-> :SYSTEM:OCANCEL 1

:SYSTem:POWeroff:EXECute

Function Turns the instrument off. This command also sets how the instrument is to behave when the main power switch turned on the next time.

Syntax :SYSTem:POWeroff:EXECute [<NRf>[,<Boolean>]]
<NRf> = 1 to 60

Example :SYSTEM:POWEROFF:EXECUTE

Description

- For <NRf>, set the time in seconds before the instrument is turned off. If <NRf> is omitted, the power is turned off after 5 seconds. Here, turning off the power means turning off the power switch on the front panel. Turning off the instrument while it is communicating with the controller may cause a communication error on the controller. To prevent communication errors, the following procedure is recommended.
 1. Enter the time required to close controller communication and send this command.
 2. Close communication between the controller and this instrument within the entered time period.
- For <Boolean>, set how the instrument is to behave when the main power switch turned on the next time. When <Boolean> is ON, the instrument will turn on when you turn off the power switch and then turn it back on. When <Boolean> is OFF, the instrument will remain off even when you turn off the power switch and then turn it back on. If <Boolean> is omitted, the behavior is the same as if it were set to OFF.

:SYSTem:PPOWer:STATus?

Function Queries the probe power supply status.

Syntax :SYSTem:PPOWer:STATus?

Example :SYSTEM:PPOWER:STATUS?
-> :SYSTEM:PPOWER:STATUS NORMAL

Description The response is as follows:

NORMAL: Probe power supply is normal.

WARNING: Probe power supply is overloaded.

FAILURE: Probe power supply is stopped.

:SYSTem:TOUCh

Function Sets or queries the touch panel on/off state.

Syntax :SYSTem:TOUCh {<Boolean>}
:SYSTem:TOUCh?

Example :SYSTEM:TOUCH ON
:SYSTEM:TOUCH?
-> :SYSTEM:TOUCH 1

:SYSTem:TOUT?

Function Queries all trigger output settings.

Syntax :SYSTem:TOUT?

5.31 SYSTem Group

:SYSTem:TOUT:POLarity

Function Sets or queries the trigger out polarity.

Syntax :SYSTem:TOUT:POLarity {NEGative|POSitive}

Example :SYSTem:TOUT:POLarity?

:SYSTEM:TOUT:POLARITY NEGATIVE

:SYSTEM:TOUT:POLARITY?

-> :SYSTEM:TOUT:POLARITY NEGATIVE

:SYSTem:TOUT:RTIMing

Function Sets or queries the timing for resetting the trigger output.

Syntax :SYSTem:TOUT:RTIMing {AEND|NASTart}

:SYSTem:TOUT:RTIMing?

Example :SYSTEM:TOUT:RTIMING AEND

:SYSTem:TOUT:RTIMING?

-> :SYSTEM:TOUT:RTIMING AEND

Description

- If "AEND" is specified, the output is reset at the end of waveform acquisition.
- If "NASTart" is specified, the output is reset at the start of the next waveform acquisition.

:SYSTem:TSYNchro?

Function Queries all time synchronization settings.

Syntax :SYSTem:TSYNchro?

:SYSTem:TSYNchro:IEEE1588?

Function Queries all IEEE 1588 time synchronization settings.

Syntax :SYSTem:TSYNchro:IEEE1588?

:SYSTem:TSYNchro:IEEE1588:DNUMber

Function Sets or queries the domain number of IEEE 1588 time synchronization.

Syntax :SYSTem:TSYNchro:IEEE1588:DNUMber <NRf>

:SYSTem:TSYNchro:IEEE1588:DNUMber?

<NRf> = 0 to 255

Example :SYSTEM:TSYNCHRO:IEEE1588:DNUMBER 0

:SYSTEM:TSYNCHRO:IEEE1588:DNUMBER?

-> :SYSTEM:TSYNCHRO:IEEE1588:DNUMBER 0

:SYSTem:TSYNchro:IEEE1588:LSECond

Function Sets or queries the leap second of IEEE 1588 time synchronization.

Syntax :SYSTem:TSYNchro:IEEE1588:LSECond {<NRf>}

:SYSTem:TSYNchro:IEEE1588:LSECond?

<NRf> = 0 to 255

Example :SYSTEM:TSYNCHRO:IEEE1588:LSECOND 0

:SYSTEM:TSYNCHRO:IEEE1588:LSECOND?

-> :SYSTEM:TSYNCHRO:IEEE1588:LSECOND 0

:SYSTem:TSYNchro:IEEE1588:NLAYer

Function Sets or queries the network layer of IEEE 1588 time synchronization.

Syntax :SYSTem:TSYNchro:IEEE1588:NLAYer {LAYER2|LAYER3}

:SYSTem:TSYNchro:IEEE1588:NLAYer?

Example :SYSTEM:TSYNCHRO:IEEE1588:NLAYER LAYER2

:SYSTEM:TSYNCHRO:IEEE1588:NLAYER?

-> :SYSTEM:TSYNCHRO:IEEE1588:NLAYER LAYER2

:SYSTem:TSYNchro:IEEE1588:PRIority<x>

Function Sets or queries the priority of IEEE 1588 time synchronization.

Syntax :SYSTem:TSYNchro:IEEE1588:PRIority<x> {<NRf>}

:SYSTem:TSYNchro:IEEE1588:PRIority<x>?

<NRf> = 0 to 255

<x> = 1 to 2

Example :SYSTEM:TSYNCHRO:IEEE1588:PRIORITY1 255

:SYSTEM:TSYNCHRO:IEEE1588:PRIORITY1?

-> :SYSTEM:TSYNCHRO:IEEE1588:PRIORITY1 255

:SYSTem:TSYNchro:IEEE1588:PState

Function Queries the state of IEEE 1588 time synchronization.

Syntax :SYSTem:TSYNchro:IEEE1588:PState?

Example :SYSTEM:TSYNCHRO:IEEE1588:PSTATE

-> :SYSTEM:TSYNCHRO:IEEE1588:PSTATE ON

Description The response is as follows:

OFF: The time synchronization feature is off.

INITIALIZING, FAULTY, DISABLED, LISTENING, UNCALIBRATED, SLAVE: States defined in IEEE 1588-2008

:SYSTem:TSYNchro:IEEE1588:SONLy

Function Sets or queries the Slave Only of IEEE 1588 time synchronization.

Syntax :SYSTem:TSYNchro:IEEE1588:
SONLy {<Boolean>}
:SYSTem:TSYNchro:IEEE1588:SONLy?

Example :SYSTEM:TSYNCHRO:IEEE1588:SONLY ON
:SYSTEM:TSYNCHRO:IEEE1588:STATE?
-> :SYSTEM:TSYNCHRO:IEEE1588:SONLY 1

:SYSTem:TSYNchro:IEEE1588:STATe

Function Sets or queries the on/off status of IEEE 1588 time synchronization.

Syntax :SYSTem:TSYNchro:IEEE1588:
STATe {<Boolean>}
:SYSTem:TSYNchro:IEEE1588:STATe?

Example :SYSTEM:TSYNCHRO:IEEE1588:STATE ON
:SYSTEM:TSYNCHRO:IEEE1588:STATE?
-> :SYSTEM:TSYNCHRO:IEEE1588:STATE 1

:SYSTem:TSYNchro:STATe?

Function Queries the instrument's time synchronization status.

Syntax :SYSTem:TSYNchro:STATe?

Example :SYSTEM:TSYNCHRO:STATE?
-> :SYSTEM:TSYNCHRO:STATE OFF

Description The response is as follows:
OFF: The time synchronization feature is off.
UNLock: Unable to communicate with the master.
The sampling clock is running based on the internal oscillator.
LOCK: Starting to synchronize with the master and adjusting the time and sampling clock.
STABle: Established synchronization with the master and the time and sampling clock are stable.

:SYSTem:USBKeyboard

Function Sets or queries the USB keyboard type.

Syntax :SYSTem:USBKeyboard {ENGLish|
JAPANEse}
:SYSTem:USBKeyboard?

Example :SYSTEM:USBKEYBOARD JAPANESE
:SYSTEM:USBKEYBOARD?
-> :SYSTEM:USBKEYBOARD JAPANESE

5.32 TImebase Group

:TImebase?

Function Queries all time base settings.

Syntax :TImebase?

:TImebase:SRATe? (Sample RATE)

Function Queries the sample rate.

Syntax :TImebase:SRATe?

Example :TIMEBASE:SRATE?

-> :TIMEBASE:SRATE 12.50E+06

:TImebase:TDIV

Function Sets or queries the Time/div value.

Syntax :TImebase:TDIV {<Time>}

:TImebase:TDIV?

<Time> = 1 ns to 500 s

Example :TIMEBASE:TDIV 2NS

:TIMEBASE:TDIV?

-> :TIMEBASE:TDIV 2.000E-09

5.33 TRIGger Group

The following table indicates how the communication commands for trigger types correspond to the trigger types that appear when you press a panel key.

Trigger Type	Communication Command	Setup Menu Panel Key	Trigger Type
Edge	SIMPlE	EDGE Key	—
OR of multiple edge triggers (edge OR)	OR	ENHANCED key	Edge OR
Multiple input combination (pattern)	PATtern		Pattern
Pulse width	PULSe		Pulse Width
Rise/fall time	RFTime		Rise/Fall Time
Runt	RUNT		Runt
Timeout value	TIMEout		Timeout
Window	WINDow		Window
Window OR	WOR		Window OR
Edge interval trigger	INTerval		Interval
Serial trigger	CAN*		CAN
	CANFD*		CAN FD
	CXPI*		CXPI
	FLEXray*		FlexRay
	I2C*		I2C
	LIN*		LIN
	PSI5*		PSI5 Airbag
	SENT*		SENT
	SPATtern		User Define
	SPI*		SPI
	UART*		UART
TV	TV:HDTV		HDTV
	TV:NTSC		NTSC
	TV:PAL		PAL
	TV:SDTV		SDTV(480/60p)
	TV:USERdefine		UserDef TV
Combination	COMBination	B TRIG key	—
Forced	FORCE	SHIFT+B TRIG (FORCE TRIG) Key	—

* CAN, CANFD, CXPI, FLEXray, I2C, LIN, PSI5, SENT, SPI, and UART are serial bus options. These serial bus commands cannot be used on models without these options.

:TRIGger?

Function Queries all trigger settings.

Syntax :TRIGger?

:TRIGger:ABN?

Function Queries all A->B(N) trigger settings.

Syntax :TRIGger:ABN?

:TRIGger:ABN:COUNT

Function Sets or queries the number of times condition B must be met for A->B(N) triggers.

Syntax :TRIGger:ABN:COUNT {<NRf>}
:TRIGger:ABN:COUNT?
<NRf> = 1 to 1000000000

Example :TRIGGER:ABN:COUNT 1
:TRIGGER:ABN:COUNT?
-> :TRIGGER:ABN:COUNT 1

:TRIGger:ACTion?

Function Queries all action-on-trigger settings.

Syntax :TRIGger:ACTion?

:TRIGger:ACTion:ACQCount

Function Sets or queries the action count of the action-on-trigger.

Syntax :TRIGger:ACTion:ACQCount {<NRf>|
INFinite}
:TRIGger:ACTion:ACQCount?
<NRf> = 1 to 1000000

Example :TRIGGER:ACTION:ACQCOUNT 1
:TRIGGER:ACTION:ACQCOUNT?
-> :TRIGGER:ACTION:ACQCOUNT 1

5.33 TRIGger Group

:TRIGger:ACTion:BUZZer

Function Sets or queries whether a warning sound is generated as an action.

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 or queries whether screen capture is output as an action.

Syntax :TRIGger:ACTion:HCOPY {<Boolean>}
:TRIGger:ACTion:HCOPY?

Example :TRIGGER:ACTION:HCOPY ON
:TRIGGER:ACTION:HCOPY?
-> :TRIGGER:ACTION:HCOPY 1

:TRIGger:ACTion:MAIL?

Function Queries all settings related to the mail notification on action.

Syntax :TRIGger:ACTion:MAIL?

:TRIGger:ACTion:MAIL:COUNT

Function Sets or queries the upper limit to the number of mail notifications for the mail notification on action.

Syntax :TRIGger:ACTion:MAIL:COUNT {<Nrf>}
:TRIGger:ACTion:MAIL:COUNT?
<Nrf> = 1 to 1000

Example :TRIGGER:ACTION:MAIL:COUNT 1
:TRIGGER:ACTION:MAIL:COUNT?
-> :TRIGGER:ACTION:MAIL:COUNT 1

:TRIGger:ACTion:MAIL:MODE

Function Sets or queries whether a mail notification is made as an action.

Syntax :TRIGger:ACTion:MAIL:
MODE {<Boolean>}
:TRIGger:ACTion:MAIL:MODE?

Example :TRIGGER:ACTION:MAIL:MODE ON
:TRIGGER:ACTION:MAIL:MODE?
-> :TRIGGER:ACTION:MAIL:MODE 1

:TRIGger:ACTion:SAVE

Function Sets or queries whether waveform data is saved to the storage device as an action.

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 action-on-trigger.

Syntax :TRIGger:ACTion:START
:TRIGGER:ACTION:START

Description To abort the trigger action, use
":TRIGger:ACTION:STOP".

:TRIGger:ACTion:STOP

Function Stops action-on-trigger.

Syntax :TRIGger:ACTion:STOP
:TRIGGER:ACTION:STOP

Description This command only stops the trigger action. It does not cancel the setting.
Use ":TRIGger:ACTION:START" to resume the trigger action.

:TRIGger:ADB?

Function Queries all A Delay B trigger settings.

Syntax :TRIGger:ADB?

:TRIGger:ADB:DElay

Function Sets or queries the delay time for condition B for A Delay B triggers.

Syntax :TRIGger:ADB:DElay {<Time>}
:TRIGger:ADB:DElay?
<Time> = 10 ns to 10 s (2 ns steps)

Example :TRIGGER:ADB:DELAY 10ns
:TRIGGER:ADB:DELAY?
-> :TRIGGER:ADB:DELAY 10ns

:TRIGger:{ATRigger|BTRigger}?

Function Queries all trigger condition settings.

Syntax :TRIGger:{ATRigger|BTRigger}?

:TRIGger{[:ATRigger]|:BTRigger}:CAN?

Function Queries all CAN bus signal trigger settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN?

:TRIGger{[:ATRigger]][:BTRigger]:CAN:BRATe

Function Sets or queries the CAN bus signal trigger bit rate (data transfer rate).

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN:
BRATe {<NRf>|USER,<NRf>}
:TRIGger{[:ATRigger]][:BTRigger]:CAN:
BRATe?
<NRf> = 33300, 83300, 125000, 250000,
500000, 1000000
USER <NRf> = See the Features Guide for this
information.

Example :TRIGGER:ATRIGGER:CAN:BRATE 83300
:TRIGGER:ATRIGGER:CAN:BRATE?
-> :TRIGGER:ATRIGGER:CAN:BRATE 83300

:TRIGger{[:ATRigger]][:BTRigger]:CAN:EFRame?

Function Queries all CAN bus signal trigger error settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN:
EFRame?

:TRIGger{[:ATRigger]][:BTRigger]:CAN:EFRame:CRC

Function Sets or queries the CAN bus signal trigger CRC error setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN:
EFRame:CRC {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:CAN:
EFRame:CRC?

Example :TRIGGER:ATRIGGER:CAN:EFRAME:CRC ON
:TRIGGER:ATRIGGER:CAN:EFRAME:CRC?
-> :TRIGGER:ATRIGGER:CAN:EFRAME:
CRC 1

:TRIGger{[:ATRigger]][:BTRigger]:CAN:EFRame[:MODE]

Function Sets or queries the CAN bus signal trigger error frame setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN:
EFRame[:MODE] {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:CAN:
EFRame:MODE?

Example :TRIGGER:ATRIGGER:CAN:EFRAME:MODE ON
:TRIGGER:ATRIGGER:CAN:EFRAME:MODE?
-> :TRIGGER:ATRIGGER:CAN:EFRAME:
MODE 1

:TRIGger{[:ATRigger]][:BTRigger]:CAN:EFRame:STUFF

Function Sets or queries the CAN bus signal trigger stuff error setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN:
EFRame:STUFF {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:CAN:
EFRame:STUFF?

Example :TRIGGER:ATRIGGER:CAN:EFRAME:
STUFF ON
:TRIGGER:ATRIGGER:CAN:EFRAME:STUFF?
-> :TRIGGER:ATRIGGER:CAN:EFRAME:
STUFF 1

:TRIGger{[:ATRigger]][:BTRigger]:CAN:IDData?

Function Queries all ID and data condition settings for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN:
IDData?

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:IDData]:ACK?

Function Queries the ACK setting, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CAN[:IDData]:ACK?

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:IDData]:ACK:MODE

Function Sets or queries the ACK mode, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CAN[:IDData]:ACK:MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
CAN[:IDData]:ACK:MODE?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:ACK:
MODE ON
:TRIGGER:ATRIGGER:CAN:IDDATA:ACK:
MODE?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:ACK:
MODE 1

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:ID Data]:ACK:TYPE

Function Sets or queries the ACK condition, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CAN[:IDData]:ACK:TYPE {ACK|ACKBoth|
NONack}
:TRIGger{[:ATRigger]][:BTRigger]:
CAN[:IDData]:ACK:TYPE?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:ACK:
TYPE ACK
:TRIGGER:ATRIGGER:CAN:IDDATA:ACK:
TYPE?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:ACK:
TYPE ACK

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:ID Data]:DATA?

Function Queries the data, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CAN[:IDData]:DATA?

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:ID Data]:DATA:CONDition

Function Sets or queries the data comparison condition, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CAN[:IDData]:DATA:CONDition {EQual|
FALSE|GREater|INSide|LESS|NOTEqual|
OUTSide|TRUE}
:TRIGger{[:ATRigger]][:BTRigger]:
CAN[:IDData]:DATA:CONDition?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
CONDITION EQUAL
:TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
CONDITION?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
DATA:CONDITION EQUAL

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:ID Data]:DATA:DECimal<x>

Function Sets a reference value, which is one of the ID and data conditions, for CAN bus signal triggering in decimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CAN[:IDData]:DATA:DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:
CAN[:IDData]:DATA:DECimal<x>?
<x> = 1 to 2
<Nrf> = See the Features Guide for this
information.

Example :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
DECIMAL1 1
:TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
DECIMAL1?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
DATA:DECIMAL1 1.000E+00

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:ID Data]:DATA:DLC

Function Sets or queries the number of significant bytes (DLC), which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CAN[:IDData]:DATA:DLC {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:
CAN[:IDData]:DATA:DLC?
<Nrf> = 0 to 8

Example :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
DLC 0
:TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
DLC?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
DATA:DLC 0

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:ID Data]:DATA:DPATtern?

Function Queries all judgment data settings for the ID and data conditions for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CAN[:IDData]:DATA:DPATtern?

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:ID Data]:DATA:DPATtern:HEXa

Function Sets a reference value, which is one of the ID and data conditions, for CAN bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CAN[:IDData]:DATA:DPATtern:
HEXa {<String>}

Example :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
DPATTERN:HEXa "12"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]][:BTRigger]:CAN[:IDData]:DATA:DLC setting.

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:ID Data]:DATA:DPATtern:PATtern

Function Sets or queries the data, which is one of the ID and data conditions, for CAN bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:DPATtern:
PATtern {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:DPATtern:PATtern?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
DPATTERN:PATTERN "00110101"
:TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
DPATTERN:PATTERN?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
DATA:DPATTERN:PATTERN "00110101"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:DATA:DLC setting.

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:ID Data]:DATA:ENDian

Function Sets or queries the byte order of the reference values, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:ENDian {BIG|
LITTLE}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:ENDian?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
ENDIAN BIG
:TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
ENDIAN?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
DATA:ENDIAN BIG

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:ID Data]:DATA:MODE

Function Sets or queries the data condition (enabled/disabled), which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:MODE?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
MODE ON
:TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
MODE?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
DATA:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:ID Data]:DATA:MSBLsb

Function Sets or queries the MSB and LSB bits for the data, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:MSBLsb {<Nrf>,
<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:MSBLsb?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
MSBLSB 1,0
:TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
MSBLSB?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
DATA:MSBLSB 1,0

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:ID Data]:DATA:SIGN

Function Sets or queries the data sign, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:SIGN {SIGN|UNSign}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:DATA:SIGN?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
SIGN SIGN
:TRIGGER:ATRIGGER:CAN:IDDATA:DATA:
SIGN?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
DATA:SIGN SIGN

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:ID Data]:IDENTifier?

Function Queries the identifier, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier?

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:ID Data]:IDENTifier:ID?

Function Queries the ID, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:ID?

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:ID Data]:IDENTifier:ID:HEXa

Function Sets an ID, which is one of the ID and data conditions, for CAN bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:ID:
HEXa{<String>}

Example :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:ID:HEXa "1AB"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:IDENTifier:MFO Rmat setting.

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:ID Data]:IDENTifier:ID:MODE

Function Sets or queries the ID condition (enabled/disabled), which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:ID:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:ID<x>:MODE?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:ID:MODE ON
:TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:ID:MODE?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:ID:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:ID Data]:IDENTifier:ID:PATtern

Function Sets or queries the ID condition, which is one of the ID and data conditions, for CAN bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:ID:
PATtern {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:ID:PATtern?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:ID:PATTERN "11100001111"
:TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:ID:PATTERN?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:ID:PATTERN "11100001111"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:CAN[:IDData]:IDENTifier:MFO Rmat setting.

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:ID Data]:IDENTifier:MFOFormat

Function Sets or queries the ID frame format (standard or extended), which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:
MFOFormat {EXTended|STANdard}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:MFOFormat?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:MFORMAT EXTENDED
:TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:MFORMAT?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:MFORMAT EXTENDED

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:ID Data]:IDENTifier:PFOFormat

Function Sets or queries the ID input format, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:
PFOFormat {MESSAge|PATtern}
:TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:IDENTifier:PFOFormat?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:PFORMAT PATTERN
:TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:PFORMAT?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:
IDENTIFIER:PFORMAT PATTERN

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:ID Data]:MSIGnal?

Function Queries the message signal, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:MSIGnal?

:TRIGger{[:ATRigger]|:BTRigger}:CAN[:ID Data]:MSIGnal:MESSAge:ITEM

Function Sets the message item, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CAN[:IDData]:MSIGnal:MESSAge:
ITEM {<String>}
<String> = Up to 32 characters

Example :TRIGGER:ATRIGGER:CAN:IDDATA:
MSIGNAL:MESSAGE:ITEM "TEST"

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:ID Data]:MSIGNAL:SElect

Function Sets or queries the CAN bus signal trigger message signal setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN[:IDData]:MSIGNAL:SElect {MESSAge|SIGNAl}
:TRIGger{[:ATRigger]][:BTRigger]:CAN[:IDData]:MSIGNAL:SElect?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SELECT MESSAGE
:TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SELECT?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SELECT MESSAGE

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:ID Data]:MSIGNAL:SIGNAL?

Function Queries the signal, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN[:IDData]:MSIGNAL:SIGNAL?

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:ID Data]:MSIGNAL:SIGNAL:CONDition

Function Sets or queries the signal data condition, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN[:IDData]:MSIGNAL:SIGNAL:CONDition {EQUAl|GREATER|INSide|LESS|NOTEQUAl|OUTSide}
:TRIGger{[:ATRigger]][:BTRigger]:CAN[:IDData]:MSIGNAL:SIGNAL:CONDition?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SIGNAL:CONDITION EQUAL
:TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SIGNAL:CONDITION?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SIGNAL:CONDITION EQUAL

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:ID Data]:MSIGNAL:SIGNAL:DECimal<x>

Function Sets a reference signal value, which is one of the ID and data conditions, for CAN bus signal triggering in decimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN[:IDData]:MSIGNAL:SIGNAL:DECimal<x>{<NRf>}
:TRIGger{[:ATRigger]][:BTRigger]:CAN[:IDData]:MSIGNAL:SIGNAL:DECimal<x>?

<x> = 1 to 2

<NRf> = See the Features Guide for this information.

Example :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SIGNAL:DECIMAL1 1
:TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SIGNAL:DECIMAL1?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SIGNAL:DECIMAL1 1.000E+00

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:ID Data]:MSIGNAL:SIGNAL:ITEM

Function Sets the signal item, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN[:IDData]:MSIGNAL:SIGNAL:ITEM {<String>,<String>}
<String> = Up to 32 characters

Example :TRIGGER:ATRIGGER:CAN:IDDATA:MSIGNAL:SIGNAL:ITEM "Sig _ Test",
"Mess _ Test"

Description Set the signal item first and then the message item.

:TRIGger{[:ATRigger]][:BTRigger]:CAN[:ID Data]:RTR

Function Sets or queries the RTR value, which is one of the ID and data conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN[:IDData]:RTR {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:CAN[:IDData]:RTR?

Example :TRIGGER:ATRIGGER:CAN:IDDATA:RTR ON
:TRIGGER:ATRIGGER:CAN:IDDATA:RTR?
-> :TRIGGER:ATRIGGER:CAN:IDDATA:RTR 1

:TRIGger{[:ATRigger]][:BTRigger]:CAN:ID OR?

Function Queries all ID OR condition settings for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN:IDOR?

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK?

Function Queries all ACK settings for the ID OR conditions for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK?

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK:MODE

Function Sets or queries the ACK mode, which is one of the ID OR conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK:MODE?

Example :TRIGGER:ATRIGGER:CAN:IDOR:ACK:MODE ON
:TRIGGER:ATRIGGER:CAN:IDOR:ACK:MODE?
-> :TRIGGER:ATRIGGER:CAN:IDOR:ACK:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK:TYPE

Function Sets or queries the ACK condition, which is one of the ID OR conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK:TYPE {ACK|ACKBoth|NONack}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:ACK:TYPE?

Example :TRIGGER:ATRIGGER:CAN:IDOR:ACK:TYPE ACK
:TRIGGER:ATRIGGER:CAN:IDOR:ACK:TYPE?
-> :TRIGGER:ATRIGGER:CAN:IDOR:ACK:TYPE ACK

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:DATA?

Function Queries all data settings for the ID OR conditions for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:DATA?

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:DATA[:MODE]

Function Sets or queries the data enable/disable condition, which is one of the ID OR conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:DATA[:MODE] {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:DATA[:MODE]?

Example :TRIGGER:ATRIGGER:CAN:IDOR:DATA:MODE ON
:TRIGGER:ATRIGGER:CAN:IDOR:DATA:MODE?
-> :TRIGGER:ATRIGGER:CAN:IDOR:DATA:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier?

Function Queries all identifier settings for the ID OR conditions for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier?

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:ID<x>?

Function Queries all ID settings for the ID OR conditions for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:ID<x>?
<x> = 1 to 4

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:ID<x>:HEXa

Function Sets the ID value, which is one of the ID OR conditions, for CAN bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:ID<x>:HEXa {<String>}
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:ID1:HEXa "1AB"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:MFormat setting.

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:ID<x>:MODE

Function Sets or queries the ID enable/disable condition, which is one of the ID OR conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:ID<x>:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:ID<x>:MODE?

Example :TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:ID1:MODE ON
:TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:ID1:MODE?
-> :TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:ID1:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:ID<x>:PATTERN

Function Sets or queries the ID condition, which is one of the ID OR conditions, for CAN bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:ID<x>:PATTERN {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:ID<x>:PATTERN?

Example :TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:ID1:PATTERN "11100001111"
:TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:ID1:PATTERN?
-> :TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:ID1:PATTERN "11100001111"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:MFORMAT setting.

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:MFORMAT

Function Sets or queries the ID frame format (standard or extended), which is one of the ID OR conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:MFORMAT {EXTENDED|STANDARD}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:MFORMAT?

Example :TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:MFORMAT EXTENDED
:TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:MFORMAT?
-> :TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:MFORMAT EXTENDED

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:MODE

Function Sets or queries the identifier enable/disable condition, which is one of the ID OR conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:MODE {<Boolean>}

Example :TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:MODE ON
-> :TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:PFORMAT

Function Sets or queries the ID input format, which is one of the ID OR conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:PFORMAT {MESSAGE|PATTERN}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:IDENTifier:PFORMAT?

Example :TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:PFORMAT PATTERN
:TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:PFORMAT?
-> :TRIGGER:ATRIGGER:CAN:IDOR:IDENTIFIER:PFORMAT PATTERN

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:MSIGNAL<x>:MESSAGE:ITEM

Function Sets the message item, which is one of the ID OR conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:MSIGNAL<x>:MESSAGE:ITEM {<String>}
<x> = 1 to 4

<String> = Up to 32 characters
Example :TRIGGER:ATRIGGER:CAN:IDOR:MSIGNAL1:MESSAGE:ITEM "TEST"

:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:RTR

Function Sets or queries the RTR, which is one of the ID OR conditions, for CAN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:RTR {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:CAN:IDOR:RTR?

Example :TRIGGER:ATRIGGER:CAN:IDOR:RTR ON
:TRIGGER:ATRIGGER:CAN:IDOR:RTR?
-> :TRIGGER:ATRIGGER:CAN:IDOR:RTR 1

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:CAN:MODE

Function Sets or queries the CAN bus signal trigger mode.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN:
MODE {EFrame|IDData|IDOR|SOF}
:TRIGger{[:ATRigger]][:BTRigger]:CAN:
MODE?

Example :TRIGGER:ATRIGGER:CAN:MODE EFRAME
:TRIGGER:ATRIGGER:CAN:MODE?
-> :TRIGGER:ATRIGGER:CAN:MODE EFRAME

:TRIGger{[:ATRigger]][:BTRigger]:CAN:RECESSIVE

Function Sets or queries the CAN bus signal trigger recessive level.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN:
RECESSIVE {HIGH|LOW}
:TRIGger{[:ATRigger]][:BTRigger]:CAN:
RECESSIVE?

Example :TRIGGER:ATRIGGER:CAN:RECESSIVE HIGH
:TRIGGER:ATRIGGER:CAN:RECESSIVE?
-> :TRIGGER:ATRIGGER:CAN:
RECESSIVE HIGH

:TRIGger{[:ATRigger]][:BTRigger]:CAN:SOURCe

Function Sets or queries the CAN bus signal trigger signal.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN:
SOURce {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:CAN:
SOURce?
<Nrf> = 1 to 4

Example :TRIGGER:ATRIGGER:CAN:SOURCe 1
:TRIGGER:ATRIGGER:CAN:SOURCe?
-> :TRIGGER:ATRIGGER:CAN:SOURCe 1

:TRIGger{[:ATRigger]][:BTRigger]:CAN:SPOint

Function Sets or queries the CAN bus signal trigger sample point.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:CAN:
SPOint {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:CAN:
SPOint?
<Nrf> = 18.8 to 90.6

Example :TRIGGER:ATRIGGER:CAN:SPOint 18.8
:TRIGGER:ATRIGGER:CAN:SPOint?
-> :TRIGGER:ATRIGGER:CAN:SPOint 18.8

:TRIGger{[:ATRigger]][:BTRigger]:CANFD?

Function Queries all CAN FD bus signal trigger settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD?

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:BRATE

Function Sets or queries the CAN FD bus signal trigger bit rate (data transfer rate).

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:BRATE {<Nrf>|USER,<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD:BRATE?

<Nrf> = 250000, 500000, 1000000
USER <Nrf> = 20000 to 1000000 (resolution:
100)

Example :TRIGGER:ATRIGGER:CANFD:BRATE 500000
:TRIGGER:ATRIGGER:CANFD:BRATE?
-> :TRIGGER:ATRIGGER:CANFD:
BRATE 500000

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:DBRate

Function Sets or queries the CAN FD bus signal trigger data phase bit rate (data transfer rate).

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:DBRate {<Nrf>|USER,<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD:DBRate?

<Nrf> = 500000, 1000000, 2000000, 4000000,
5000000, 8000000
USER <Nrf> = 250000 to 10000000, resolution:
100

Example :TRIGGER:ATRIGGER:CANFD:
DBRATE 1000000
:TRIGGER:ATRIGGER:CANFD:DBRATE?
-> :TRIGGER:ATRIGGER:CANFD:
DBRATE 1000000

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:DSPoint

Function Sets or queries the CAN FD bus signal trigger data phase sample point.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:DSPoint {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD:DSPoint?

<Nrf> = 18.8 to 90.6, resolution: 0.1

Example :TRIGGER:ATRIGGER:CANFD:DSPoint 18.8
:TRIGGER:ATRIGGER:CANFD:DSPoint?
-> :TRIGGER:ATRIGGER:CANFD:
DSPoint 18.8

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFrFrame?

Function Queries all CAN FD bus signal trigger error settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrFrame?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFrFrame:CRC

Function Sets or queries the CAN FD bus signal trigger CRC error setting.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrFrame:CRC {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrFrame:CRC?

Example :TRIGGER:ATRIGGER:CANFD:EFrFRAME:
CRC ON
:TRIGGER:ATRIGGER:CANFD:EFrFRAME:CRC?
-> :TRIGGER:ATRIGGER:CANFD:EFrFRAME:
CRC 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFrFrame:CRCEFactor?

Function Queries all CRC error factor settings for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrFrame:CRCEFactor?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFrFrame:CRCEFactor:CRCSequence

Function Sets or queries the CRC sequence (a CRC error factor) for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrFrame:CRCEFactor:
CRCSequence {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrFrame:CRCEFactor:CRCSequence?

Example :TRIGGER:ATRIGGER:CANFD:EFrFRAME:
CRCEFACTOR:CRCSEQUENCE ON
:TRIGGER:ATRIGGER:CANFD:EFrFRAME:
CRCEFACTOR:CRCSEQUENCE?
-> :TRIGGER:ATRIGGER:CANFD:EFrFRAME:
CRCEFACTOR:CRCSEQUENCE 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFrFrame:CRCEFactor:SCount

Function Sets or queries the Stuff Count (a CRC error factor) for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrFrame:CRCEFactor:
SCount {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrFrame:CRCEFactor:SCount?

Example :TRIGGER:ATRIGGER:CANFD:EFrFRAME:
CRCEFACTOR:SCOUNT ON
:TRIGGER:ATRIGGER:CANFD:EFrFRAME:
CRCEFACTOR:SCOUNT?
-> :TRIGGER:ATRIGGER:CANFD:EFrFRAME:
CRCEFACTOR:SCOUNT 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFrFrame:FSTuff

Function Sets or queries the CAN FD bus signal trigger fixed stuff error setting.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrFrame:FSTuff {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrFrame:FSTuff?

Example :TRIGGER:ATRIGGER:CANFD:EFrFRAME:
FSTUFF ON
:TRIGGER:ATRIGGER:CANFD:EFrFRAME:
FSTUFF?
-> :TRIGGER:ATRIGGER:CANFD:EFrFRAME:
FSTUFF 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:EFrFrame[:MODE]

Function Sets or queries the CAN FD bus signal trigger error frame setting.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrFrame[:MODE] {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:EFrFrame:mode?

Example :TRIGGER:ATRIGGER:CANFD:EFrFRAME:
MODE ON
:TRIGGER:ATRIGGER:CANFD:EFrFRAME:MODE?
-> :TRIGGER:ATRIGGER:CANFD:EFrFRAME:
MODE 1

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:EFrAmE:STUFF

Function Sets or queries the CAN FD bus signal trigger stuff error setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:EFrAmE:STUFF {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD:EFrAmE:STUFF?

Example :TRIGGER:ATRIGGER:CANFD:EFrAmE:
STUFF ON
:TRIGGER:ATRIGGER:CANFD:EFrAmE:
STUFF?
-> :TRIGGER:ATRIGGER:CANFD:EFrAmE:
STUFF 1

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:FDF?

Function Queries all CAN FD bus signal trigger FDF settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:FDF?

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:FDF:CONDition

Function Sets or queries the CAN FD bus signal trigger FDF.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:FDF:CONDition {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD:FDF:CONDition?

Example :TRIGGER:ATRIGGER:CANFD:EFrAmE:FDF:
CONDition ON
:TRIGGER:ATRIGGER:CANFD:EFrAmE:FDF:
CONDition?
-> :TRIGGER:ATRIGGER:CANFD:EFrAmE:
FDF:CONDition 1

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:FDSTandard

Function Sets or queries whether the CAN FD bus signal for triggering is an ISO standard signal.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:FDSTandard {ISO|NISO}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD:FDSTandard?

Example :TRIGger:ATRIGGER:CANFD:
FDSTANDARD ISO
:TRIGger:ATRIGGER:CANFD:FDSTANDARD?
-> :TRIGger:ATRIGGER:CANFD:
FDSTANDARD ISO

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDDatA?

Function Queries all ID and data condition settings for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDDatA?

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDDatA]:ACK?

Function Queries all ACK settings for the ID and data conditions for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDDatA]:ACK?

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDDatA]:ACK:MODE

Function Sets or queries the ACK mode, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDDatA]:ACK:MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDDatA]:ACK:MODE?

Example :TRIGGER:ATRIGGER:CANFD:IDDatA]:ACK:
MODE ON
:TRIGGER:ATRIGGER:CANFD:IDDatA]:ACK:
MODE?
-> :TRIGGER:ATRIGGER:CANFD:IDDatA]:
ACK:MODE 1

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDDatA]:ACK:TYPE

Function Sets or queries the ACK condition, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDDatA]:ACK:TYPE {ACK|
ACKBoth|NONack}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDDatA]:ACK:TYPE?

Example :TRIGGER:ATRIGGER:CANFD:IDDatA]:ACK:
TYPE ACK
:TRIGGER:ATRIGGER:CANFD:IDDatA]:ACK:
TYPE?
-> :TRIGGER:ATRIGGER:CANFD:IDDatA]:
ACK:TYPE ACK

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDDatA]:DATA?

Function Queries all data settings for the ID and data conditions for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDDatA]:DATA?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:BCOUNT

Function Sets or queries the pattern comparison position, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:BCOUNT {<Nrf>}
<Nrf> = 0 to 63

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
BCOUNT 4
:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
BCOUNT?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
DATA:BCOUNT 4

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:CONDition

Function Sets or queries the comparison condition, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:
CONDition {EQual|FALSE|GREater|
INSide|LESS|NOTEqual|OUTSide|TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:CONDition?

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
CONDITION EQUAL
:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
CONDITION?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
DATA:CONDITION EQUAL

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:DBYTe

Function Sets or queries the number of data bytes, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:DBYTe {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:DBYTe?
<Nrf> = 0 to 8

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
DBYTE 2
:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
DBYTE?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
DATA:DBYTE 2

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:DECimal<x>

Function Sets or queries a reference values (a or b), which is one of the ID and data conditions, for CAN FD bus signal triggering in decimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:
DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:DECimal<x>?
<x> = 1 to 2
<Nrf> = -9E+18 to 9E+18

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
DECIMAL1 1
:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
DECIMAL1?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
DATA:DECIMAL1 1.000E+00

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:DPATtern?

Function Queries all judgment data settings for the ID and data conditions for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:DPATtern?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:DPATtern:HEXa

Function Sets a reference value, which is one of the ID and data conditions, for CAN FD bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:DATA:DPATtern:
HEXa {<String>}

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
DPATTERN:HEXA "12"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:DATA:DBYTe setting.

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:CANFD[:IDData]:DATA:DPATtern:PATtern

Function Sets or queries the data value, which is one of the ID and data conditions, for CAN FD bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:DATA:DPATtern:
PATtern {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:DATA:DPATtern:
PATtern?

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
DPATTERN:PATTERN "00110101"
:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
DPATTERN:PATTERN?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
DATA:DPATTERN:PATTERN "00110101"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]][:BTRigger]:CANFD[:IDData]:DATA:DBYte setting.

:TRIGger{[:ATRigger]][:BTRigger]:CANFD[:IDData]:DATA:ENDian

Function Sets or queries the byte order of the reference values, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:DATA:ENDian {BIG|
LITTLE}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:DATA:ENDian?

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
ENDIAN BIG
:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
ENDIAN?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
DATA:ENDIAN BIG

:TRIGger{[:ATRigger]][:BTRigger]:CANFD[:IDData]:DATA:MODE

Function Sets or queries the data enable/disable condition, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:DATA:MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:DATA:MODE?

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
MODE ON
:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
MODE?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
DATA:MODE 1

:TRIGger{[:ATRigger]][:BTRigger]:CANFD[:IDData]:DATA:MSBLsb

Function Sets or queries the MSB and LSB bits for the data, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:DATA:MSBLsb {<Nrf>,
<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:DATA:MSBLsb?
<Nrf> = 0 to 63

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
MSBLSB 1,0
:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
MSBLSB?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
DATA:MSBLSB 1,0

:TRIGger{[:ATRigger]][:BTRigger]:CANFD[:IDData]:DATA:SIGN

Function Sets or queries whether signed or unsigned data format will be used, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:DATA:SIGN {SIGN|
UNSign}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:DATA:SIGN?

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
SIGN SIGN
:TRIGGER:ATRIGGER:CANFD:IDDATA:DATA:
SIGN?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
DATA:SIGN SIGN

:TRIGger{[:ATRigger]][:BTRigger]:CANFD[:IDData]:IDENTifier?

Function Queries all identifier settings for the ID and data conditions for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:IDENTifier?

:TRIGger{[:ATRigger]][:BTRigger]:CANFD[:IDData]:IDENTifier:ID?

Function Queries all identifier settings for the ID and data conditions for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:IDENTifier:ID?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDeNtifier:ID:HEXa

Function Sets the ID value, which is one of the ID and data conditions, for CAN FD bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDeNtifier:ID:
HEXa {<String>}

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:ID:HEXA "1AB"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDeNtifier:MFORMAT setting.

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDeNtifier:ID:MODE

Function Sets or queries the ID enable/disable condition, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDeNtifier:ID:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDeNtifier:ID<x>:
MODE?

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:ID:MODE ON
:TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:ID:MODE?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:ID:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDeNtifier:ID:PATteRn

Function Sets or queries the ID condition, which is one of the ID and data conditions, for CAN FD bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDeNtifier:ID:
PATteRn {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDeNtifier:ID:
PATteRn?

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:ID:PATTERN "11100001111"
:TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:ID:PATTERN?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:ID:PATTERN "11100001111"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDeNtifier:MFORMAT setting.

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDeNtifier:MFORMAT

Function Sets or queries the ID frame format (standard or extended), which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDeNtifier:
MFORMAT {EXTended|STANdard}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDeNtifier:MFORMAT?

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:MFORMAT EXTENDED
:TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:MFORMAT?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:MFORMAT EXTENDED

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:IDeNtifier:PFORMAT

Function Sets or queries the ID input format, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDeNtifier:
PFORMAT {MESSAge|PATteRn}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:IDeNtifier:PFORMAT?

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:PFORMAT PATTERN
:TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:PFORMAT?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
IDENTIFIER:PFORMAT PATTERN

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:MSIGnaL?

Function Queries all message and signal settings for the ID and data conditions for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:MSIGnaL?

:TRIGger{[:ATRigger]|:BTRigger}:CANFD[:IDData]:MSIGnaL:MESSAge:ITEM

Function Sets the message item, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD[:IDData]:MSIGnaL:MESSAge:
ITEM {<String>}
<String> = Up to 32 characters

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:MESSAGE:ITEM "TEST"

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:CANFD[:IDData]:MSIGnal:SElect

Function Sets or queries the message and signal condition, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:MSIGnal:
SElect {MESSAge|SIGNal}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:MSIGnal:SElect?

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SELECT MESSAGE
:TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SELECT?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SELECT MESSAGE

:TRIGger{[:ATRigger]][:BTRigger]:CANFD[:IDData]:MSIGnal:SIGNAL?

Function Queries all signal settings for the ID and data conditions for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:MSIGnal:SIGNAL?

:TRIGger{[:ATRigger]][:BTRigger]:CANFD[:IDData]:MSIGnal:SIGNAL:CONDition

Function Sets or queries the signal condition, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:MSIGnal:SIGNAL:
CONDition {EQUal|GREater|INSide|
LESS|NOTEqual|OUTSide}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:MSIGnal:SIGNAL:
CONDition?

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SIGNAL:CONDITION EQUAL
:TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SIGNAL:CONDITION?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SIGNAL:CONDITION EQUAL

:TRIGger{[:ATRigger]][:BTRigger]:CANFD[:IDData]:MSIGnal:SIGNAL:DECimal<x>

Function Sets or queries a signal's reference value, which is one of the ID and data conditions, for CAN FD bus signal triggering in decimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:MSIGnal:SIGNAL:
DECimal<x> {<NRf>}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:MSIGnal:SIGNAL:
DECimal<x>?

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SIGNAL:DECIMAL1 1
:TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SIGNAL:DECIMAL1?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SIGNAL:DECIMAL1 1.000E+00

:TRIGger{[:ATRigger]][:BTRigger]:CANFD[:IDData]:MSIGnal:SIGNAL:ITEM

Function Sets the signal item, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:MSIGnal:SIGNAL:
ITEM {<String>,<String>}
<String> = Up to 32 characters

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:
MSIGNAL:SIGNAL:ITEM "Sig _ Test",
"Mess _ Test"

Description Set the signal item first and then the message item.

:TRIGger{[:ATRigger]][:BTRigger]:CANFD[:IDData]:RTR

Function Sets or queries the RTR, which is one of the ID and data conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:RTR {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD[:IDData]:RTR?

Example :TRIGGER:ATRIGGER:CANFD:IDDATA:
RTR ON
:TRIGGER:ATRIGGER:CANFD:IDDATA:RTR?
-> :TRIGGER:ATRIGGER:CANFD:IDDATA:
RTR 1

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDOR?

Function Queries all ID OR condition settings for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDOR?

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDOR:ACK?

Function Queries all ACK settings for the ID OR conditions for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDOR:ACK?

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDOR:ACK:MODE

Function Sets or queries the ACK mode, which is one of the ID OR conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDOR:ACK:MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDOR:ACK:MODE?

Example :TRIGGER:ATRIGGER:CANFD:IDOR:ACK:
MODE ON
:TRIGGER:ATRIGGER:CANFD:IDOR:ACK:
MODE?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:ACK:
MODE 1

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDOR:ACK:TYPE

Function Sets or queries the ACK condition, which is one of the ID OR conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDOR:ACK:TYPE {ACK|ACKBoth|
NONack}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDOR:ACK:TYPE?

Example :TRIGGER:ATRIGGER:CANFD:IDOR:ACK:
TYPE ACK
:TRIGGER:ATRIGGER:CANFD:IDOR:ACK:TYPE?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:ACK:
TYPE ACK

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDOR:DATA?

Function Queries all data settings for the ID OR conditions for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDOR:DATA?

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDOR:DATA[:MODE]

Function Sets or queries the data enable/disable condition, which is one of the ID OR conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDOR:DATA[:MODE] {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDOR:DATA[:MODE]?

Example :TRIGGER:ATRIGGER:CANFD:IDOR:DATA:
MODE ON
:TRIGGER:ATRIGGER:CANFD:IDOR:DATA:
MODE?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:
DATA:MODE 1

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDOR:IDENTifier?

Function Queries all identifier settings for the ID OR conditions for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDOR:IDENTifier?

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDOR:IDENTifier:ID<x>?

Function Queries all ID settings for the ID OR conditions for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDOR:IDENTifier:ID<x>?
<x> = 1 to 4

:TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDOR:IDENTifier:ID<x>:HEXa

Function Sets the ID value, which is one of the ID OR conditions, for CAN FD bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CANFD:IDOR:IDENTifier:ID<x>:
HEXa {<String>}
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:ID1:HEXa "1AB"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]][:BTRigger]:CANFD:IDOR:IDENTifier:MFORM at setting.

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:ID<x>:MODE

Function Sets or queries the ID enable/disable condition, which is one of the ID OR conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:IDENTifier:ID<x>:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:IDENTifier:ID<x>:MODE?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:ID1:MODE ON
:TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:ID1:MODE?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:ID1:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:ID<x>:PATtern

Function Sets or queries the ID condition, which is one of the ID OR conditions, for CAN FD bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:IDENTifier:ID<x>:
PATtern {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:IDENTifier:ID<x>:PATtern?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:ID1:PATTERN "11100001111"
:TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:ID1:PATTERN?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:ID1:PATTERN "11100001111"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:MFORMAT setting.

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:MFORMAT

Function Sets or queries the ID frame format (standard or extended), which is one of the ID OR conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:IDENTifier:
MFORMAT {EXTended|STANdard}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:IDENTifier:MFORMAT?

Example :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:MFORMAT EXTENDED
:TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:MFORMAT?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:MFORMAT EXTENDED

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:MODE

Function Sets or queries the ID enable/disable condition, which is one of the ID OR conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:IDENTifier:
MODE {<Boolean>}

Example :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:MODE ON
-> :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:IDENTifier:PFORMAT

Function Sets or queries the ID input format, which is one of the ID OR conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:IDENTifier:
PFORMAT {MESSAge|PATtern}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:IDENTifier:PFORMAT?

Example :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:PFORMAT PATTERN
:TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:PFORMAT?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:
IDENTIFIER:PFORMAT PATTERN

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:MSIGNAL<x>:MESSAGE:ITEM

Function Sets the message item, which is one of the ID OR conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:MSIGNAL<x>:MESSAGE:
ITEM {<String>}
<x> = 1 to 4
<String> = Up to 32 characters

Example :TRIGGER:ATRIGGER:CAN:IDOR:MSIGNAL1:
MESSAGE:ITEM "TEST"

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:IDOR:RTR

Function Sets or queries the RTR, which is one of the ID OR conditions, for CAN FD bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:RTR {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:IDOR:RTR?

Example :TRIGGER:ATRIGGER:CANFD:IDOR:RTR ON
:TRIGGER:ATRIGGER:CANFD:IDOR:RTR?
-> :TRIGGER:ATRIGGER:CANFD:IDOR:
RTR 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:MODE

Function Sets or queries the CAN FD bus signal trigger mode.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:MODE {EFrame|ESI|FDF|IDData|
IDOR|SOF}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:MODE?

Example :TRIGGER:ATRIGGER:CANFD:MODE EFRAME
:TRIGGER:ATRIGGER:CANFD:MODE?
-> :TRIGGER:ATRIGGER:CANFD:
MODE EFRAME

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:RECESSIVE

Function Sets or queries the CAN FD bus signal trigger recessive level.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:RECESSIVE {HIGH|LOW}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:RECESSIVE?

Example :TRIGGER:ATRIGGER:CANFD:
RECESSIVE HIGH
:TRIGGER:ATRIGGER:CANFD:RECESSIVE?
-> :TRIGGER:ATRIGGER:CANFD:
RECESSIVE HIGH

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:SOURce

Function Sets or queries the CAN FD bus signal trigger signal.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:SOURce {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:SOURce?
<Nrf> = 1 to 4

Example :TRIGGER:ATRIGGER:CANFD:SOURCE 1
:TRIGGER:ATRIGGER:CANFD:SOURCE?
-> :TRIGGER:ATRIGGER:CANFD:SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:CANFD:SPOint

Function Sets or queries the CAN FD bus signal trigger sample point.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CANFD:SPOint {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CANFD:SPOint?

Example <Nrf> = 18.8 to 90.6 (resolution: 0.1)
:TRIGGER:ATRIGGER:CANFD:SPOINT 18.8
:TRIGGER:ATRIGGER:CANFD:SPOINT?
-> :TRIGGER:ATRIGGER:CANFD:
SPOINT 18.8

:TRIGger{[:ATRigger]|:BTRigger}:CXPI?

Function Queries all CXPI bus signal trigger settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI?

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:BRATe

Function Sets or queries the CXPI bus signal trigger bit rate (data transfer rate).

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:BRATe {<Nrf>|USER,<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:BRATe?

<Nrf> = 4800, 9600, 19200
USER <Nrf> = 4000 to 50000
Example :TRIGGER:ATRIGGER:CXPI:BRATE 19200
:TRIGGER:ATRIGGER:CXPI:BRATE?
-> :TRIGGER:ATRIGGER:CXPI:
BRATE 19200

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:CTOLerance

Function Sets or queries the CXPI bus signal trigger clock tolerance.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:CTOLerance {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:CTOLerance?

<Nrf> = $\pm 0.5\%$ to $\pm 5.0\%$, resolution: 0.1%
Example :TRIGGER:ATRIGGER:CXPI:
CTOLERANCE 0.04
:TRIGGER:ATRIGGER:CXPI:CTOLERANCE?
-> :TRIGGER:ATRIGGER:CXPI:
CTOLERANCE 0.04

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ERROr?

Function Queries all CXPI bus signal trigger error settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERROr?

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ER Ror:CLOCK

Function Sets or queries the CXPI bus signal trigger clock error setting.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERror:CLOCK {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERror:CLOCK?

Example :TRIGGER:ATRIGGER:CXPI:ERROR:
CLOCK ON
:TRIGGER:ATRIGGER:CXPI:ERROR:CLOCK?
-> :TRIGGER:ATRIGGER:CXPI:ERROR:
CLOCK 1

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ER Ror:CRC

Function Sets or queries the CXPI bus signal trigger CRC error setting.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERror:CRC {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERror:CRC?

Example :TRIGGER:ATRIGGER:CXPI:ERROR:CRC ON
:TRIGGER:ATRIGGER:CXPI:ERROR:CRC?
-> :TRIGGER:ATRIGGER:CXPI:ERROR:
CRC 1

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ER Ror:DLEnGth

Function Sets or queries the CXPI bus signal trigger data length error setting.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERror:DLEnGth {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERror:DLEnGth?

Example :TRIGGER:ATRIGGER:CXPI:ERROR:
DLENGTH ON
:TRIGGER:ATRIGGER:CXPI:ERROR:
DLENGTH?
-> :TRIGGER:ATRIGGER:CXPI:ERROR:
DLENGTH 1

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ER Ror:FRAMing

Function Sets or queries the CXPI bus signal trigger framing error setting.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERror:FRAMing {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERror:FRAMing?

Example :TRIGGER:ATRIGGER:CXPI:ERROR:
FRAMING ON
:TRIGGER:ATRIGGER:CXPI:ERROR:
FRAMING?
-> :TRIGGER:ATRIGGER:CXPI:ERROR:
FRAMING 1

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ER Ror:IBS

Function Sets or queries the CXPI bus signal trigger IBS error setting.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERror:IBS {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERror:IBS?

Example :TRIGGER:ATRIGGER:CXPI:ERROR:IBS ON
:TRIGGER:ATRIGGER:CXPI:ERROR:IBS?
-> :TRIGGER:ATRIGGER:CXPI:ERROR:
IBS 1

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ER Ror:PARity

Function Sets or queries the CXPI bus signal trigger parity error setting.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERror:PARity {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:ERror:PARity?

Example :TRIGGER:ATRIGGER:CXPI:ERROR:
PARITY ON
:TRIGGER:ATRIGGER:CXPI:ERROR:PARITY?
-> :TRIGGER:ATRIGGER:CXPI:ERROR:
PARITY 1

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID Data?

Function Queries all ID and data condition settings for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData?

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID
Data:DATA?**

Function Queries the data, which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA?

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID
Data:DATA:BCount**

Function Sets or queries the pattern comparison position, which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:BCount {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:BCount?
<Nrf> = 0 to 254

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
BCOUNT 1
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
BCOUNT?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:BCOUNT 1

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID
Data:DATA:CONDition**

Function Sets or queries the data comparison condition, which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:CONDition {EQUAL|
FALSE|GREATER|INSIDE|LESS|NOTEQUAL|
OUTSIDE|TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:CONDition?

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
CONDITION EQUAL
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
CONDITION?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:CONDITION EQUAL

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID
Data:DATA:DBYTE**

Function Sets or queries the number of data bytes, which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:DBYTE {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:DBYTE?
<Nrf> = 0 to 8

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
DBYTE 1
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
DBYTE?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:DBYTE 1

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID
Data:DATA:DECimal<x>**

Function Sets a reference value, which is one of the ID and data conditions, for CXPI bus signal triggering in decimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:DECimal<x>?
<x> = 1 to 2
<Nrf> = -9E+18 to 9E+18

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
DECIMAL1 1
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
DECIMAL1?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:DECIMAL1 1.0000000E+00

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID
Data:DATA:DPATtern?**

Function Queries the ID, which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:DPATtern?

**:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID
Data:DATA:DPATtern:HEXa**

Function Sets an ID, which is one of the ID and data conditions, for CXPI bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:DATA:DPATtern:
HEXa {<String>}

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
DPATTERN:HEXA "12"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:CXPI:IDData:DATA:DBYTE setting.

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:ID Data:DATA:DPATtern:PATtern

Function Sets or queries the ID condition, which is one of the ID and data conditions, for CXPI bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:DATA:DPATtern:
PATtern {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:DATA:DPATtern:PATtern?

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
DPATTERN:PATTERN "00110101"
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
DPATTERN:PATTERN?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:DPATTERN:PATTERN "00110101"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]][:BTRigger]:CXPI:IDData:DATA:DBYTE setting.

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:ID Data:DATA:ENDian

Function Sets or queries the byte order of the reference values, which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:DATA:ENDian {BIG|LITTLE}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:DATA:ENDian?

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
ENDIAN BIG
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
ENDIAN?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:ENDIAN BIG

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:ID Data:DATA:MODE

Function Sets or queries the ID and data condition (enabled/disabled) for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:DATA:MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:DATA:MODE?

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
MODE ON
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
MODE?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:MODE 1

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:ID Data:DATA:MSBLsb

Function Sets or queries the MSB and LSB bits for the data, which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:DATA:MSBLsb {<Nrf>,
<Nrf>}
<Nrf> = 0 to 63

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
MSLSB 7,0
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
MSLSB?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:MSLSB 7,0

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:ID Data:DATA:SIGN

Function Sets or queries the data sign, which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:DATA:SIGN {SIGN|UNSign}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:DATA:SIGN?

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
SIGN SIGN
:TRIGGER:ATRIGGER:CXPI:IDDATA:DATA:
SIGN?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
DATA:SIGN SIGN

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:ID Data:FINformation?

Function Queries all frame information settings for the ID and data conditions for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:FINformation?

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:ID Data:FINformation:CT

Function Sets or queries the frame information counter value, which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:FINformation:CT {<Nrf>|
DONTcare}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:FINformation:CT?
<Nrf> = 0 to 3

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:CT 3
:TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:CT?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:CT 3

**:TRIGger{[:ATRigger]][:BTRigger]:CXPI:ID
Data:FINformation:MODE**

Function Sets or queries the frame information (enabled/disabled), which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:FINformation:
MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:FINformation:MODE?

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:MODE ON
:TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:MODE?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:MODE 1

**:TRIGger{[:ATRigger]][:BTRigger]:CXPI:ID
Data:FINformation:SLEEP**

Function Sets or queries the frame information sleep bit, which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:FINformation:SLEEP {0|1|
X}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:FINformation:SLEEP?

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:SLEEP 1
:TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:SLEEP?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:SLEEP 1

**:TRIGger{[:ATRigger]][:BTRigger]:CXPI:ID
Data:FINformation:WAKEup**

Function Sets or queries the frame information wakeup bit, which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:FINformation:WAKEup {0|
1|X}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:FINformation:WAKEup

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:WAKEUP 1
:TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:WAKEUP?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:
FINFOMATION:WAKEUP 1

**:TRIGger{[:ATRigger]][:BTRigger]:CXPI:ID
Data:ID?**

Function Queries the ID, which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:ID?

**:TRIGger{[:ATRigger]][:BTRigger]:CXPI:ID
Data:ID:HEXa**

Function Sets an ID, which is one of the ID and data conditions, for CXPI bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:ID:HEXa {<String>}

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
HEXa "1E"

**:TRIGger{[:ATRigger]][:BTRigger]:CXPI:ID
Data:ID:MODE**

Function Sets or queries the ID condition (enabled/disabled), which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:ID:MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:ID:MODE?

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
MODE ON
:TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
MODE?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
MODE 1

**:TRIGger{[:ATRigger]][:BTRigger]:CXPI:ID
Data:ID:PATtern**

Function Sets or queries the ID, which is one of the ID and data conditions, for CXPI bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:ID:PATtern {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDData:ID:PATtern?

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
PATTERN "0010000"
:TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
PATTERN?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
PATTERN "0010000"

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID Data:ID:PTYPE

Function Sets or queries the PTYPE comparison condition, which is one of the ID and data conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:ID:PTYPE {DONTcare|NO|
YES}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDData:ID:PTYPE?

Example :TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
PTYPYE DONTCARE
:TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
PTYPYE?
-> :TRIGGER:ATRIGGER:CXPI:IDDATA:ID:
PTYPYE DONTCARE

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID OR?

Function Queries all ID OR condition settings for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR?

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID OR:IDENTifier?

Function Queries all identifier settings for the IDOR conditions for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier?

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID OR:IDENTifier:ID<x>?

Function Queries all ID settings for the IDOR conditions for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:ID<x>?
<x> = 1 to 4

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID OR:IDENTifier:ID<x>:HEXa

Function Sets the ID value, which is one of the IDOR conditions, for CXPI bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:ID<x>:
HEXa {<String>}
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:ID1:HEXA "1E"

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID OR:IDENTifier:ID<x>:MODE

Function Sets or queries the ID enable/disable condition, which is one of the IDOR conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:ID<x>:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:ID<x>:MODE?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:ID1:MODE ON
:TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:ID1:MODE?
-> :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:ID1:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID OR:IDENTifier:ID<x>:PATTERN

Function Sets or queries the ID condition, which is one of the IDOR conditions, for CXPI bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:ID<x>:
PATTERN {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:ID<x>:PATTERN?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:ID1:PATTERN "0010000"
:TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:ID1:PATTERN?
-> :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:ID1:PATTERN "0010000"

:TRIGger{[:ATRigger]|:BTRigger}:CXPI:ID OR:IDENTifier:MODE

Function Sets or queries the identifier enable/disable condition, which is one of the IDOR conditions, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
CXPI:IDOR:IDENTifier:MODE?

Example :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:MODE ON
:TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:MODE?
-> :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:MODE 1

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:IDOR:IDENTifier:PTYPE

Function Sets or queries the comparison condition of PTYPE, which is an IDOR condition, for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDOR:IDENTifier:
PTYPE {DONTcare|NO|YES}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:IDOR:IDENTifier:PTYPE?

Example :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:PTYPE DONTCARE
:TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:PTYPE?
-> :TRIGGER:ATRIGGER:CXPI:IDOR:
IDENTIFIER:PTYPE DONTCARE

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:MODE

Function Sets or queries the CXPI bus signal trigger type.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:MODE {ERROR|IDData|IDOR|PTYPE|
SOF|WAKEupsleep}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:MODE?

Example :TRIGGER:ATRIGGER:CXPI:MODE SOF
:TRIGGER:ATRIGGER:CXPI:MODE?
-> :TRIGGER:ATRIGGER:CXPI:MODE SOF

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:PTYPE?

Function Queries all CXPI bus signal trigger PTYPE settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:PTYPE?

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:PTYPE:CONDition

Function Sets or queries the CXPI bus signal trigger PTYPE condition setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:PTYPE:CONDition {EPTYPE|
NOResponse}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:PTYPE:CONDition?

Example :TRIGGER:ATRIGGER:CXPI:PTYPE:
CONDITION EPTYPE
:TRIGGER:ATRIGGER:CXPI:PTYPE:
CONDITION?
-> :TRIGGER:ATRIGGER:CXPI:PTYPE:
CONDITION EPTYPE

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:SOURce

Function Sets or queries the CXPI bus signal trigger signal.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:SOURce {<NRF>}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:SOURce?
<NRF> = 1 to 4

Example :TRIGGER:ATRIGGER:CXPI:SOURCE 1
:TRIGGER:ATRIGGER:CXPI:SOURCE?
-> :TRIGGER:ATRIGGER:CXPI:SOURCE 1

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:TSAMPLE

Function Sets or queries the logic value (1 or 0) determination threshold for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:TSAMPLE <NRF>
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:TSAMPLE?
<NRF> = 0.01 Tbit to 0.3 Tbit, resolution:
0.001 Tbit

Example :TRIGGER:ATRIGGER:CXPI:TSAMPLE 0.04
:TRIGGER:ATRIGGER:CXPI:TSAMPLE?
-> :TRIGGER:ATRIGGER:CXPI:
TSAMPLE 0.04

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:WAKEupsleep?

Function Queries all wakeup and sleep settings for CXPI bus signal triggering.

:TRIGger{[:ATRigger]][:BTRigger]:CXPI:WAKEupsleep:TYPE

Function Sets or queries the wakeup and sleep conditions for CXPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
CXPI:WAKEupsleep:TYPE {SFrame|SLEEP|
WAKEup|WPULse}
:TRIGger{[:ATRigger]][:BTRigger]:
CXPI:WAKEupsleep:TYPE?

Example :TRIGGER:ATRIGGER:CXPI:WAKEUPSLEEP:
TYPE SFRAME
:TRIGGER:ATRIGGER:CXPI:WAKEUPSLEEP:
TYPE?
-> :TRIGGER:ATRIGGER:CXPI:
WAKEUPSLEEP:TYPE SFRAME

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray?

Function Queries all FlexRay bus signal trigger settings.
Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray?

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray:BCHannel

Function Sets or queries the channel bus type for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:BCHannel {A|B}
:TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:BCHannel?

Example :TRIGGER:ATRIGGER:FLEXRAY:BCHANNEL A
:TRIGGER:ATRIGGER:FLEXRAY:BCHANNEL?
-> :TRIGGER:ATRIGGER:FLEXRAY:
BCHANNEL A

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray:BRATe

Function Sets or queries the FlexRay bus signal trigger bit rate (data transfer rate).

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:BRATe {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:BRATe?

<Nrf> = 2500000, 5000000, 10000000

Example :TRIGGER:ATRIGGER:FLEXRAY:
BRATE 5000000
:TRIGGER:ATRIGGER:FLEXRAY:BRATE?
-> :TRIGGER:ATRIGGER:FLEXRAY:
BRATE 5000000

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray:ERRor?

Function Queries all FlexRay bus signal trigger error settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:ERRor?

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray:ERRor:BSS

Function Sets or queries the FlexRay bus signal trigger BSS error setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:ERRor:BSS {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:ERRor:BSS?

Example :TRIGGER:ATRIGGER:FLEXRAY:ERROR:
BSS ON
:TRIGGER:ATRIGGER:FLEXRAY:ERROR:BSS?
-> :TRIGGER:ATRIGGER:FLEXRAY:ERROR:
BSS 1

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray:ERRor:CRC

Function Sets or queries the FlexRay bus signal trigger CRC error setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:ERRor:CRC {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:ERRor:CRC?

Example :TRIGGER:ATRIGGER:FLEXRAY:ERROR:
CRC ON
:TRIGGER:ATRIGGER:FLEXRAY:ERROR:CRC?
-> :TRIGGER:ATRIGGER:FLEXRAY:ERROR:
CRC 1

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray:ERRor:FES

Function Sets or queries the FlexRay bus signal trigger FES error setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:ERRor:FES {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:ERRor:FES?

Example :TRIGGER:ATRIGGER:FLEXRAY:ERROR:
FES ON
:TRIGGER:ATRIGGER:FLEXRAY:ERROR:FES?
-> :TRIGGER:ATRIGGER:FLEXRAY:ERROR:
FES 1

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray:IDData?

Function Queries all ID and data condition settings for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:IDData?

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCount?

Function Queries the cycle count, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCount?

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCount:CONDition

Function Sets or queries the cycle count data condition, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCount:
CONDition {EQUAL|GREATER|INSIDE|
LESS|NOTEQUAL|OUTSIDE}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCount:CONDition?

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:CONDITION EQUAL
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:CONDITION?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:CONDITION EQUAL

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCount:COUNT<x>

Function Sets or queries the cycle count, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCount:
COUNT<x> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCount:COUNT<x>?
<x> = 1 to 2
<NRF> = 0 to 63

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:COUNT1 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:COUNT1?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:COUNT1 1

Description • **When** :TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCount:CONDition {EQUAL|GREATER|NOTEQUAL} **is specified**, **use** :TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCount:COUNT1 **to set the value**.
• **When** :TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCount:CONDition LESS **is specified**, **use** :TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCOUNT:COUNT2 **to set the value**.
• **When** :TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCount:CONDition {BETWEEN|NOTBETWEEN} **is specified**, **use** :TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCOUNT:COUNT1 **to set the larger value and** :TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCOUNT:COUNT2 **to set the smaller value**.

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:CCount:MODE

Function Sets or queries the cycle count condition (enabled/disabled), which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCount:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:CCount:MODE?

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:MODE ON
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:MODE?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
CCOUNT:MODE 1

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray[:IDData]:DATA<x>?

Function Queries the data field, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray[:IDData]:DATA<x>?
<x> = 1 to 2

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray[:IDData]:DATA<x>:BCount

Function Sets or queries the pattern comparison position of the data field data, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray[:IDData]:DATA<x>:
BCount {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:
FLEXray[:IDData]:DATA<x>:BCount?
<x> = 1 to 2
<Nrf> = 0 to 253

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:BCOUNT 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:BCOUNT?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:BCOUNT 1

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray[:IDData]:DATA<x>:CONDition

Function Sets or queries the data field data condition, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray[:IDData]:DATA<x>:
CONDition {EQUAL|FALSE|GREATER|
INSIDE|LESS|NOTEQUAL|OUTSIDE|TRUE}
:TRIGger{[:ATRigger]][:BTRigger]:
FLEXray[:IDData]:DATA<x>:CONDition?
<x> = 1 to 2

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:CONDITION EQUAL
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:CONDITION?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:CONDITION EQUAL

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray[:IDData]:DATA<x>:DBYTE

Function Sets or queries the number of bytes of the data field, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray[:IDData]:DATA<x>:
DBYTE {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:
FLEXray[:IDData]:DATA<x>:DBYTE?
<x> = 1 to 2
<Nrf> = 1 to 8

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:DBYTE 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:DBYTE?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:DBYTE 1

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray[:IDData]:DATA<x>:DECimal<y>

Function Sets a reference value, which is one of the ID and data conditions, for FlexRay bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray[:IDData]:DATA<x>:
DECimal<y> {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:
FLEXray[:IDData]:DATA<x>:DECimal<y>?
<x> = 1 to 2
<y> = 1 to 2
<Nrf> = See the Features Guide for this information.

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:DECIMAL1 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:DECIMAL1?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:DECIMAL1 1.000E+00

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray[:IDData]:DATA<x>:DPATtern?

Function Queries all data field judgment data settings for the ID and data conditions for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray[:IDData]:DATA:DPATtern?
<x> = 1 to 2

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:DPATtern:HEXA

Function Sets a reference value for the data field, which is one of the ID and data conditions, for FlexRay bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA:DPATtern:
HEXA {<String>}
<x> = 1 to 2

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:DPATTERN:HEXA "12"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:DBYte setting.

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:DPATtern:PATtern

Function Sets or queries the data field data, which is one of the ID and data conditions, for FlexRay bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA:DPATtern:
PATtern {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA:DPATtern:
PATtern?
<x> = 1 to 2

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:DPATTERN:PATTERN "00110101"
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:DPATTERN:PATTERN?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:DPATTERN:PATTERN "00110101"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:DBYte setting.

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:ENDian

Function Sets or queries the data field endian, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:
ENDian {BIG|LITTLE}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:ENDian?
<x> = 1 to 2

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:ENDIAN BIG
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:ENDIAN?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:ENDIAN BIG

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:MODE

Function Sets or queries the data field condition (enabled/disabled), which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:MODE?
<x> = 1 to 2

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:MODE ON
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:MODE?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:MSBLsb

Function Sets or queries the MSB/LSB bit of the data field data, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:
MSBLsb {<Nrf>,<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:MSBLsb?
<x> = 1 to 2

<Nrf> = See the Features Guide for this information.

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:MSBLSB 7,0
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:MSBLSB?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA1:MSBLSB 7,0

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:DATA<x>:SIGN

Function Sets or queries the data sign of the data field, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:SIGN {SIGN|
UNSign}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:DATA<x>:SIGN?
<x> = 1 to 2

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:SIGN SIGN
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:SIGN?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
DATA:SIGN SIGN

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:FID?

Function Queries the frame ID, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:FID?

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:FID:CONDition

Function Sets or queries the frame ID data condition, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:FID:
CONDition {EQUAL|GREATER|INSide|
LESS|NOTEQUAL|OUTSide}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:FID:CONDition?

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:CONDITION EQUAL
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:CONDITION?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:CONDITION EQUAL

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:FID:ID<x>

Function Sets or queries the frame ID value, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:FID:ID<x> {<NRF>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:FID:ID<x>?
<x> = 1 to 2
<NRF> = 1 to 2047

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:ID1 100
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:ID1?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:ID1 100

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:FID:MODE

Function Sets or queries the frame ID condition (enabled/disabled), which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:FID:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:FID:MODE?

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:MODE ON
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:MODE?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
FID:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:INDicator?

Function Queries the indicator, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator?

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:INDicator:MODE

Function Sets or queries the indicator condition (enabled/disabled), which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:MODE?

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:MODE ON
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:MODE?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:INDicator:NFrame

Function Sets or queries the indicator null frame, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:
NFrame {0|1|X}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:NFrame?

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:NFRAME 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:NFRAME?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:NFRAME 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:INDicator:PPreamble

Function Sets or queries the indicator payload preamble, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:
PPreamble {0|1|X}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:
PPreamble?

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:PPREAMBLE 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:PPREAMBLE?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:PPREAMBLE 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:INDicator:STFrame

Function Sets or queries the indicator start frame, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:
STFrame {0|1|X}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:STFrame?

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:STFRAME 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:STFRAME?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:STFRAME 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray[:IDData]:INDicator:SYFrame

Function Sets or queries the indicator sync frame, which is one of the ID and data conditions, for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:
SYFrame {0|1|X}
:TRIGger{[:ATRigger]|:BTRigger}:
FLEXray[:IDData]:INDicator:SYFrame?

Example :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:SYFRAME 1
:TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:SYFRAME?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDDATA:
INDICATOR:SYFRAME 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:IDOR?

Function Queries all OR condition settings for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR?

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:IDOR:ID<x>?

Function Queries all settings related to the specified frame ID of the OR condition for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>?
<x> = 1 to 4

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray:IDOR:ID<x>:CCount?

Function Queries all settings related to the cycle count of the specified frame ID of the OR condition for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
FLEXray:IDOR:ID<x>:CCount?
<x> = 1 to 4

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray :IDOR:ID<x>:CCOUNT:CONDition

Function Sets or queries the cycle count of the specified frame ID of the OR condition for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger):
FLEXray:IDOR:ID<x>:CCOUNT:
CONDition {DONTcare|EQUAL|GREATER|
INSide|LESS|NOTEQUAL|OUTSide}
:TRIGger{[:ATRigger]|:BTRigger):
FLEXray:IDOR:ID<x>:CCOUNT:CONDition?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
CCOUNT:CONDITION EQUAL
:TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
CCOUNT:CONDITION?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDOR:
ID1:CCOUNT:CONDITION EQUAL

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray :IDOR:ID<x>:CCOUNT:COUNT<y>

Function Sets or queries the reference cycle count value of the specified frame ID of the OR condition for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger):
FLEXray:IDOR:ID<x>:CCOUNT:
COUNT<y> {<NRF>}
:TRIGger{[:ATRigger]|:BTRigger):
FLEXray:IDOR:ID<x>:CCOUNT:COUNT<y>?
<x> = 1 to 4
<y> = 1 to 2
<NRF> = 0 to 63

Example :TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
CCOUNT:COUNT1 100
:TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
CCOUNT:COUNT1?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDOR:
ID1:CCOUNT:COUNT1 100

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray :IDOR:ID<x>:FID?

Function Queries all settings related to the ID condition of the specified frame ID of the OR condition for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger):
FLEXray:IDOR:ID<x>:FID?
<x> = 1 to 4

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray :IDOR:ID<x>:FID:CONDition

Function Sets or queries the ID judgment condition of the specified frame ID of the OR condition for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger):
FLEXray:IDOR:ID<x>:FID:
CONDition {EQUAL|GREATER|INSide|
LESS|NOTEQUAL|OUTSide}
:TRIGger{[:ATRigger]|:BTRigger):
FLEXray:IDOR:ID<x>:FID:CONDition?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
FID:CONDITION EQUAL
:TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
FID:CONDITION?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDOR:
ID1:FID:CONDITION EQUAL

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray :IDOR:ID<x>:FID:ID<y>

Function Sets or queries the ID reference value of the specified frame ID of the OR condition for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger):
FLEXray:IDOR:ID<x>:FID:ID<y> {<NRF>}
:TRIGger{[:ATRigger]|:BTRigger):
FLEXray:IDOR:ID<x>:FID:ID<y>?
<x> = 1 to 4
<y> = 1 to 2
<NRF> = 1 to 2047

Example :TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
FID:ID1 1
:TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
FID:ID1?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDOR:
ID1:FID:ID1 1

:TRIGger{[:ATRigger]|:BTRigger}:FLEXray :IDOR:ID<x>:MODE

Function Sets or queries the specified frame ID condition (enabled/disabled) of the OR condition for FlexRay bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger):
FLEXray:IDOR:ID<x>:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger):
FLEXray:IDOR:ID<x>:MODE?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
MODE ON
:TRIGGER:ATRIGGER:FLEXRAY:IDOR:ID1:
MODE?
-> :TRIGGER:ATRIGGER:FLEXRAY:IDOR:
ID1:MODE 1

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray:MODE

Function Sets or queries the FlexRay bus signal trigger mode.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:MODE {Error|FStart|IDData|
IDOR}
:TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:MODE?

Example :TRIGGER:ATRIGGER:FLEXRAY:
MODE FSTART
:TRIGGER:ATRIGGER:FLEXRAY:MODE?
-> :TRIGGER:ATRIGGER:FLEXRAY:
MODE FSTART

:TRIGger{[:ATRigger]][:BTRigger]:FLEXray:SOURce

Function Sets or queries the FlexRay bus signal trigger source.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:SOURce {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:
FLEXray:SOURce?
<Nrf> = 1 to 4

Example :TRIGGER:ATRIGGER:FLEXRAY:SOURCE 1
:TRIGGER:ATRIGGER:FLEXRAY:SOURCE?
-> :TRIGGER:ATRIGGER:FLEXRAY:
SOURCE 1

:TRIGger{[:ATRigger]][:BTRigger]:I2C?

Function Queries all I2C bus signal trigger settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C?

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADDRESS?

Function Queries all I2C bus signal trigger address pattern settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS?

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADDRESS:ADDRESS?

Function Queries all settings related to the address of the I2C bus signal trigger address pattern.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS?

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADDRESS:ADDRESS:BIT10ADDRESS?

Function Queries all I2C bus signal trigger 10-bit address settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:BIT10ADDRESS?

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADDRESS:ADDRESS:BIT10ADDRESS:DIRection

Function Sets or queries the 10-bit address data direction for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:BIT10ADDRESS:
DIRection {READ|WRITE|X}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:BIT10ADDRESS:
DIRection?

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT10ADDRESS:DIRECTION READ
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT10ADDRESS:DIRECTION?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT10ADDRESS:DIRECTION READ

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADDRESS:ADDRESS:BIT10ADDRESS:HEXa

Function Sets the 10-bit address for I2C bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:BIT10ADDRESS:
HEXa {<String>}

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT10ADDRESS:HEXa "1AB"

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADDRESS:ADDRESS:BIT10ADDRESS:PATtern

Function Sets or queries the 10-bit address for I2C bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:BIT10ADDRESS:
PATtern {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:BIT10ADDRESS:
PATtern?

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT10ADDRESS:
PATTERN "11010111001"
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT10ADDRESS:PATTERN?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT10ADDRESS:
PATTERN "11010111001"

5.33 TRIGger Group

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:ADD
Ress:ADDRESS:BIT7Address?**

Function Queries all I2C bus signal trigger 7-bit address settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7Address?

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:ADD
Ress:ADDRESS[:BIT7Address]:DIREction**

Function Sets or queries the 7-bit address data direction for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS[:BIT7Address]:
DIREction {READ|WRITE|X}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS[:BIT7Address]:
DIREction?

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7ADDRESS:DIRECTION READ
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7ADDRESS:DIRECTION?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7ADDRESS:DIRECTION READ

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:ADD
Ress:ADDRESS[:BIT7Address]:HEXa**

Function Sets the 7-bit address for I2C bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7Address:
HEXa {<String>}

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7ADDRESS:HEXA "5C"

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:ADD
Ress:ADDRESS[:BIT7Address]:PATtern**

Function Sets or queries the 7-bit address for I2C bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS[:BIT7Address]:
PATtern {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS[:BIT7Address]:
PATtern?

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7ADDRESS:
PATTERN "11100110"
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7ADDRESS:PATTERN?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7ADDRESS:
PATTERN "11100110"

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:ADD
Ress:ADDRESS:BIT7APsub?**

Function Queries all I2C bus signal trigger 7-bit + sub address settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub?

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:ADD
Ress:ADDRESS:BIT7APsub:ADDRESS?**

Function Queries all settings related to the 7-bit address of the 7-bit + sub address for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS?

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:ADD
Ress:ADDRESS:BIT7APsub:ADDRESS:DIREcti
on**

Function Sets or queries the 7-bit + sub address data direction for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
DIREction {READ|WRITE|X}
:TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
DIREction?

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:ADDRESS:
DIRECTION READ
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:ADDRESS:DIRECTION?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:ADDRESS:
DIRECTION READ

**:TRIGger{[:ATRigger]|:BTRigger}:I2C:ADD
Ress:ADDRESS:BIT7APsub:ADDRESS:HEXa**

Function Sets the 7-bit address of 7-bit + sub address for I2C bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
HEXa {<String>}

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:ADDRESS:HEXA "AB"

**:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADD
Ress:ADDRESS:BIT7APsub:ADDRESS:PATtern**

Function Sets or queries the 7-bit address of 7-bit + sub address for I2C bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
PATtern {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:BIT7APsub:ADDRESS:
PATtern?

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:ADDRESS:
PATTERN "00111010"
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:ADDRESS:PATTERN?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:ADDRESS:
PATTERN "00111010"

**:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADD
Ress:ADDRESS:BIT7APsub:SADDRESS?**

Function Queries all settings related to the sub address of the 7-bit + sub address for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS?

**:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADD
Ress:ADDRESS:BIT7APsub:SADDRESS:HEXa**

Function Sets the sub address of 7-bit + sub address for I2C bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS:
HEXa {<String>}

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:SADDRESS:HEXA "EF"

**:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADD
Ress:ADDRESS:BIT7APsub:SADDRESS:PATte
rn**

Function Sets or queries the sub address of 7-bit + sub address for I2C bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS:
PATtern {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:BIT7APsub:SADDRESS:
PATtern?

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:SADDRESS:
PATTERN "00111010"
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:SADDRESS:PATTERN?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:BIT7APSUB:SADDRESS:
PATTERN "00111010"

**:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADD
Ress:ADDRESS:MODE**

Function Sets or queries the address condition enable/disable state for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:MODE?

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:MODE ON
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:MODE?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:MODE 1

**:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADD
Ress:ADDRESS:TYPE**

Function Sets or queries the address condition address format for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:TYPE {BIT10Address|
BIT7Address|BIT7APsub}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:ADDRESS:TYPE?

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:TYPE BIT10ADDRESS
:TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:TYPE?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
ADDRESS:TYPE BIT10ADDRESS

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADD Ress:DATA?

Function Queries all settings related to the data of the I2C bus signal trigger address pattern.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:DATA?

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADD Ress:DATA:BCount

Function Sets or queries the data pattern comparison position for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:DATA:BCount {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:DATA:BCount?
<Nrf> = 0 to 9999

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
BCOUNT 0
:TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
BCOUNT?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
DATA:BCOUNT 0

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADD Ress:DATA:BMODE

Function Sets or queries on/off state of the data pattern comparison position for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:DATA:BMODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:DATA:BMODE?

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
BMODE ON
:TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
BMODE?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
DATA:BMODE 1

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADD Ress:DATA:CONDition

Function Sets or queries the data judgment method (true/false) for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:DATA:CONDition {FALSE|TRUE}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:DATA:CONDition?

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
CONDITION FALSE
:TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
CONDITION?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
DATA:CONDITION FALSE

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADD Ress:DATA:DBYTE

Function Sets or queries the number of setup data values for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:DATA:DBYTE {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:DATA:DBYTE?
<Nrf> = 1 to 4

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
DBYTE 1
:TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
DBYTE?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
DATA:DBYTE 1

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADD Ress:DATA:DPATtern?

Function Queries all I2C bus signal trigger data settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:DATA:DPATtern?

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADD Ress:DATA:DPATtern:HEXA

Function Sets the data for I2C bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:DATA:DPATtern:HEXA {<String>}

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
DPATtern:HEXA "AB"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]][:BTRigger]:I2C:ADDRESS:DATA:DBYTE setting.

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADD Ress:DATA:DPATtern:PATtern

Function Sets or queries the data condition data for I2C bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:DATA:DPATtern:
PATtern {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:
ADDRESS:DATA:DPATtern:PATtern?

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
DPATtern:PATtern "10001101"
:TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:
DPATtern:PATtern?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:
DATA:DPATtern:PATtern1 "10001101"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]][:BTRigger]:I2C:ADDRESS:DATA:DBYTE setting.

:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADDRESS:DATA:MODE

Function Sets or queries the data condition enable/disable state for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:ADDRESS:DATA:MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:ADDRESS:DATA:MODE?

Example :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:MODE ON
:TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:MODE?
-> :TRIGGER:ATRIGGER:I2C:ADDRESS:DATA:MODE 1

:TRIGger{[:ATRigger]][:BTRigger]:I2C:GENERALCALL?

Function Queries all I2C bus signal trigger general call settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:GENERALCALL?

:TRIGger{[:ATRigger]][:BTRigger]:I2C:GENERALCALL:BIT7MAddress?

Function Queries all settings related to the 7-bit master address of general call for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:GENERALCALL:BIT7MAddress?

:TRIGger{[:ATRigger]][:BTRigger]:I2C:GENERALCALL:BIT7MAddress:HEXA

Function Sets the 7-bit master address of general call for I2C bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:GENERALCALL:BIT7MAddress:HEXA {<String>}

Example :TRIGGER:ATRIGGER:I2C:GENERALCALL:BIT7MADDRESS:HEXA "AB"

:TRIGger{[:ATRigger]][:BTRigger]:I2C:GENERALCALL:BIT7MAddress:PATtern

Function Sets or queries the 7-bit master address of general call for I2C bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:GENERALCALL:BIT7MAddress:PATtern {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:GENERALCALL:BIT7MAddress:PATtern?

Example :TRIGGER:ATRIGGER:I2C:GENERALCALL:BIT7MADDRESS:PATTERN "0010110"
:TRIGGER:ATRIGGER:I2C:GENERALCALL:BIT7MADDRESS:PATTERN?
-> :TRIGGER:ATRIGGER:I2C:GENERALCALL:BIT7MADDRESS:PATTERN "0010110"

:TRIGger{[:ATRigger]][:BTRigger]:I2C:GENERALCALL:SBYTE

Function Sets or queries the general call second byte type for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:GENERALCALL:SBYTE {BIT7MAddress|DONTcare|H04|H06}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:GENERALCALL:SBYTE?

Example :TRIGGER:ATRIGGER:I2C:GENERALCALL:SBYTE BIT7MADDRESS
:TRIGGER:ATRIGGER:I2C:GENERALCALL:SBYTE?
-> :TRIGGER:ATRIGGER:I2C:GENERALCALL:SBYTE BIT7MADDRESS

:TRIGger{[:ATRigger]][:BTRigger]:I2C:INCLUDErw

Function Sets or queries the enable/disable condition of address R/W for I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:INCLUDErw {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:INCLUDErw?

Example :TRIGGER:ATRIGGER:I2C:INCLUDERW ON
:TRIGGER:ATRIGGER:I2C:INCLUDERW?
-> :TRIGGER:ATRIGGER:I2C:INCLUDERW 1

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:I2C:MODE

Function Sets or queries the I2C bus signal trigger mode.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:MODE {ADDRData|EVERystart|GENeralcall|HSMODE|NONAck|STARTbyte}:TRIGger{[:ATRigger]][:BTRigger]:I2C:MODE?

Example :TRIGGER:ATRIGGER:I2C:MODE ADDRDATA
:TRIGGER:ATRIGGER:I2C:MODE?
-> :TRIGGER:ATRIGGER:I2C:MODE ADDRDATA

:TRIGger{[:ATRigger]][:BTRigger]:I2C:NONAck?

Function Queries all I2C bus signal trigger NON ACK ignore mode settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:NONAck?

:TRIGger{[:ATRigger]][:BTRigger]:I2C:NONAck:HSMODE

Function Sets or queries whether NONACK is to be ignored in high-speed mode of I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:NONAck:HSMODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:NONAck:HSMODE?

Example :TRIGGER:ATRIGGER:I2C:NONACK:HSMODE ON
:TRIGGER:ATRIGGER:I2C:NONACK:HSMODE?
-> :TRIGGER:ATRIGGER:I2C:NONACK:HSMODE 1

:TRIGger{[:ATRigger]][:BTRigger]:I2C:NONAck:READAccess

Function Sets or queries whether NON ACK is to be ignored in the read access byte of I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:NONAck:READAccess {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:NONAck:READAccess?

Example :TRIGGER:ATRIGGER:I2C:NONACK:READACCESS ON
:TRIGGER:ATRIGGER:I2C:NONACK:READACCESS?
-> :TRIGGER:ATRIGGER:I2C:NONACK:READACCESS 1

:TRIGger{[:ATRigger]][:BTRigger]:I2C:NONAck:STARTbyte

Function Sets or queries whether NON ACK is to be ignored in the start byte of I2C bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:NONAck:STARTbyte {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:NONAck:STARTbyte?

Example :TRIGGER:ATRIGGER:I2C:NONACK:STARTBYTE ON
:TRIGGER:ATRIGGER:I2C:NONACK:STARTBYTE?
-> :TRIGGER:ATRIGGER:I2C:NONACK:STARTBYTE 1

:TRIGger{[:ATRigger]][:BTRigger]:I2C:SCL

Function Sets or queries the I2C bus signal trigger clock signal.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:SCL {<Nrf>|BIT<x>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:SCL?
<Nrf> = 1 to 4
BIT<x>'s <x> = 1 to 8

Example :TRIGGER:ATRIGGER:I2C:SCL 1
:TRIGGER:ATRIGGER:I2C:SCL?
-> :TRIGGER:ATRIGGER:I2C:SCL 1

:TRIGger{[:ATRigger]][:BTRigger]:I2C:SDA

Function Sets or queries the I2C bus signal trigger data signal.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:I2C:SDA {<Nrf>|BIT<x>}
:TRIGger{[:ATRigger]][:BTRigger]:I2C:SDA?
<Nrf> = 1 to 4
BIT<x>'s <x> = 1 to 8

Example :TRIGGER:ATRIGGER:I2C:SDA 1
:TRIGGER:ATRIGGER:I2C:SDA?
-> :TRIGGER:ATRIGGER:I2C:SDA 1

Description The data signal that you can specify varies depending on the :TRIGger{[:ATRigger]][:BTRigger]:I2C:SCL setting. For details, see the Features Guide.

:TRIGger{[:ATRigger]][:BTRigger]:LIN?

Function Queries all LIN bus signal trigger settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN?

:TRIGger{[:ATRigger]][:BTRigger]:LIN:BLENGTH

Function Sets or queries the LIN bus signal trigger break length.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:BLENGTH {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:LIN:BLENGTH?

Example <Nrf> = 10 to 13
:TRIGGER:ATRIGGER:LIN:LENGTH 10
:TRIGGER:ATRIGGER:LIN:LENGTH?
-> :TRIGGER:ATRIGGER:LIN:LENGTH 10

:TRIGger{[:ATRigger]][:BTRigger]:LIN:BRATe

Function Sets or queries the LIN bus signal trigger bit rate (data transfer rate).

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:BRATe {<Nrf>|USER,<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:LIN:BRATe?

Example <Nrf> = 1200, 2400, 4800, 9600, 19200
USER <Nrf> = See the Features Guide for this information.
:TRIGGER:ATRIGGER:LIN:BRATE 2400
:TRIGGER:ATRIGGER:LIN:BRATE?
-> :TRIGGER:ATRIGGER:LIN:BRATE 2400

:TRIGger{[:ATRigger]][:BTRigger]:LIN:ERRor?

Function Queries all LIN bus signal trigger error settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:ERRor?

:TRIGger{[:ATRigger]][:BTRigger]:LIN:ERRor:PARity

Function Sets or queries the LIN bus signal trigger parity error setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:ERRor:PARity {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:LIN:ERRor:PARity?

Example :TRIGGER:ATRIGGER:LIN:ERROR:PARITY ON
:TRIGGER:ATRIGGER:LIN:ERROR:PARITY?
-> :TRIGGER:ATRIGGER:LIN:ERROR:PARITY 1

:TRIGger{[:ATRigger]][:BTRigger]:LIN:ERRor:SYNCh

Function Sets or queries the LIN bus signal trigger synch error setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:ERRor:SYNCh {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:LIN:ERRor:SYNCh?

Example :TRIGGER:ATRIGGER:LIN:ERROR:SYNCH ON
:TRIGGER:ATRIGGER:LIN:ERROR:SYNCH?
-> :TRIGGER:ATRIGGER:LIN:ERROR:SYNCH 1

:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData?

Function Queries all ID and data settings for LIN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData?

:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:DATA?

Function Queries all LIN bus signal trigger data settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:DATA?

:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:DATA:CONDition

Function Sets or queries the LIN bus signal trigger data condition setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:DATA:CONDition {EQUAL|FALSE|GREATER|INSIDE|LESS|NOTEQUAL|OUTSIDE|TRUE}
:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:DATA:CONDition?

Example :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:CONDITION EQUAL
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:CONDITION?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:CONDITION EQUAL

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DBYTE

Function Sets or queries the number of data bytes for LIN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DBYTE {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DBYTE?
<NRf> = 1 to 8

Example :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:DBYTE 1
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:DBYTE?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:DBYTE 1

:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DECimal<x>

Function Sets or queries the data for LIN bus signal triggering in decimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DECimal<x> {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DECimal<x>?
<x> = 1 to 2
<NRf> = See the Features Guide for this information.

Example :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:DECIMAL1 1
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:DECIMAL1?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:DECIMAL1 1.000E+00

:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DPATtern?

Function Queries all LIN bus signal trigger data settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DPATtern?

:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DPATtern:HEXa

Function Sets the data for LIN bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DPATtern:HEXa {<String>}

Example :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:DPATTERN:HEXA "12"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DBYTE setting.

:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DPATtern:PATtern

Function Sets or queries the data for LIN bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DPATtern:PATtern {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DPATtern:PATtern?

Example :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:DPATTERN:PATTERN "00110101"
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:DPATTERN:PATTERN?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:DPATTERN:PATTERN "00110101"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:DBYTE setting.

:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:ENDian

Function Sets or queries the data endian for LIN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:ENDian {BIG|LITTLE}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:ENDian?

Example :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:ENDIAN BIG
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:ENDIAN?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:ENDIAN BIG

:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:MODE

Function Sets or queries the data condition enable/disable state for LIN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDData:DATA:MODE?

Example :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:MODE ON
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:MODE?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:MODE 1

:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:MSBLSb

Function Sets or queries the MSB/LSB bit for LIN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:DATA:MSBLSb {<NRf>,<NRf>}
:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:DATA:MSBLSb?
<NRf> = 0 to 63

Example :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:MSBLSB 0
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:MSBLSB?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:MSBLSB 0

:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:SIGN

Function Sets or queries the data sign for LIN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:DATA:SIGN {SIGN|UNSign}
:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:DATA:SIGN?

Example :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:SIGN SIGN
:TRIGGER:ATRIGGER:LIN:IDDATA:DATA:SIGN?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:DATA:SIGN SIGN

:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:IDENTifier?

Function Queries all LIN bus signal trigger identifier settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:IDENTifier?

:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:IDENTifier:ID?

Function Queries all LIN bus signal trigger ID settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:IDENTifier:ID?

:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:IDENTifier:ID:HEXa

Function Sets the ID for LIN bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:IDENTifier:ID:HEXa {<String>}
:TRIGGER:ATRIGGER:LIN:IDDATA:IDENTIFIER:ID:HEXA "1E"

:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:IDENTifier:ID:MODE

Function Sets or queries the ID condition (enabled/disabled), which is one of the ID and data conditions, for LIN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:IDENTifier:ID:MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:IDENTifier:ID:MODE?

Example :TRIGGER:ATRIGGER:LIN:IDDATA:IDENTIFIER:ID:MODE ON
:TRIGGER:ATRIGGER:LIN:IDDATA:IDENTIFIER:ID:MODE?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:IDENTIFIER:ID:MODE 1

:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:IDENTifier:ID:PATtern

Function Sets or queries the ID for LIN bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:IDENTifier:ID:PATtern {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDData:IDENTifier:ID:PATtern?

Example :TRIGGER:ATRIGGER:LIN:IDDATA:IDENTIFIER:ID:PATTERN "101100"
:TRIGGER:ATRIGGER:LIN:IDDATA:IDENTIFIER:ID:PATTERN?
-> :TRIGGER:ATRIGGER:LIN:IDDATA:IDENTIFIER:ID:PATTERN "101100"

:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDOR?

Function Queries all ID OR condition settings for LIN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:IDOR?

:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDOR:IDENTifier?

Function Queries all ID settings for the ID OR conditions for LIN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:IDOR:IDENTifier?

:TRIGger{[:ATRigger]][:BTRigger]:LIN:IDOR:IDENTifier:ID<x>?

Function Queries all settings related to the specified ID of the ID OR condition for LIN bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:LIN:IDOR:IDENTifier:ID<x>?
<x> = 1 to 4

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDOR:IDENTifier:ID<x>:HEXa

Function Sets the ID value, which is one of the ID OR conditions, for LIN bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:LIN:IDOR:IDENTifier:ID<x>:HEXa {<String>}
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:LIN:IDOR:IDENTIFIER:ID1:HEXA "1E"

:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDOR:IDENTifier:ID<x>:MODE

Function Sets or queries the ID enable/disable condition, which is one of the ID OR conditions, for LIN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:LIN:IDOR:IDENTifier:ID<x>:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDOR:IDENTifier:ID<x>:MODE?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:LIN:IDOR:IDENTIFIER:ID1:MODE ON
:TRIGGER:ATRIGGER:LIN:IDOR:IDENTIFIER:ID1:MODE?
-> :TRIGGER:ATRIGGER:LIN:IDOR:IDENTIFIER:ID1:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDOR:IDENTifier:ID<x>:PATtern

Function Sets or queries the ID condition, which is one of the ID OR conditions, for LIN bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:LIN:IDOR:IDENTifier:ID<x>:PATtern {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDOR:IDENTifier:ID<x>:PATtern?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:LIN:IDOR:IDENTIFIER:ID1:PATTERN "101100"
:TRIGGER:ATRIGGER:LIN:IDOR:IDENTIFIER:ID1:PATTERN?
-> :TRIGGER:ATRIGGER:LIN:IDOR:IDENTIFIER:ID1:PATTERN "101100"

:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDOR:IDENTifier:MODE

Function Sets or queries the ID condition enable/disable state for LIN bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:LIN:IDOR:IDENTifier:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:IDOR:IDENTifier:MODE?

Example :TRIGGER:ATRIGGER:LIN:IDOR:IDENTIFIER:MODE ON
:TRIGGER:ATRIGGER:LIN:IDOR:IDENTIFIER:MODE?
-> :TRIGGER:ATRIGGER:LIN:IDOR:IDENTIFIER:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:LIN:MODE

Function Sets or queries the LIN bus signal trigger mode.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:LIN:MODE {BSYNch|ERRor|IDData|IDOR}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:MODE?

Example :TRIGGER:ATRIGGER:LIN:MODE BSYNCH
:TRIGGER:ATRIGGER:LIN:MODE?
-> :TRIGGER:ATRIGGER:LIN:MODE BSYNCH

:TRIGger{[:ATRigger]|:BTRigger}:LIN:SOURce

Function Sets or queries the LIN bus signal trigger signal.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:LIN:SOURce {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:SOURce?
<Nrf> = 1 to 4

Example :TRIGGER:ATRIGGER:LIN:SOURCE 1
:TRIGGER:ATRIGGER:LIN:SOURCE?
-> :TRIGGER:ATRIGGER:LIN:SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:LIN:SPOint

Function Sets or queries the LIN bus signal trigger sample point.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:LIN:SPOint {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:LIN:SPOint?
<Nrf> = 18.8 to 90.6

Example :TRIGGER:ATRIGGER:LIN:SPOINT 18.8
:TRIGGER:ATRIGGER:LIN:SPOINT?
-> :TRIGGER:ATRIGGER:LIN:SPOINT 18.8

:TRIGger{[:ATRigger]][:BTRigger]:OR?

Function Queries all edge OR trigger settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:OR?

:TRIGger{[:ATRigger]][:BTRigger]:OR:ALL

Function Sets or queries whether to make all channels edge OR trigger sources at once.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:OR:
ALL {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:OR:
ALL?

Example :TRIGGER:ATRIGGER:OR:ALL ON
:TRIGGER:ATRIGGER:OR:ALL?
-> :TRIGGER:ATRIGGER:OR:ALL 1

:TRIGger{[:ATRigger]][:BTRigger]:OR:CHANnel<x>

Function Sets or queries the slope of each channel of the edge OR trigger.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:OR:
CHANnel<x>{BOTH|FALL|OFF|RISE}
:TRIGger{[:ATRigger]][:BTRigger]:OR:
CHANnel<x>?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:OR:CHANNEL1 RISE
:TRIGGER:ATRIGGER:OR:CHANNEL1?
-> :TRIGGER:ATRIGGER:OR:
CHANNEL1 RISE

:TRIGger{[:ATRigger]][:BTRigger]:PATTe rn?

Function Queries all settings related to the combination (pattern) trigger of multiple inputs.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PATTe rn?

:TRIGger{[:ATRigger]][:BTRigger]:PATTe rn:BITS?

Function Queries all settings related to the logic inputs of the combination (pattern) trigger of multiple inputs.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PATTe rn:BITS?

:TRIGger{[:ATRigger]][:BTRigger]:PATTe rn:BITS:HEXa

Function Sets the logic input state of the combination (pattern) trigger of multiple inputs in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PATTe rn:BITS:HEXa {<String>}

Example :TRIGGER:ATRIGGER:PATTERN:BITS:
HEXa "1F"

:TRIGger{[:ATRigger]][:BTRigger]:PATTe rn:BITS:PATTe rn

Function Sets or queries the logic input pattern of the combination (pattern) trigger of multiple inputs.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PATTe rn:BITS:PATTe rn {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:
PATTe rn:BITS:PATTe rn?

Example :TRIGGER:ATRIGGER:PATTERN:BITS:
PATTERN "10111000"
:TRIGGER:ATRIGGER:PATTERN:BITS:
PATTERN?
-> :TRIGGER:ATRIGGER:PATTERN:BITS:
PATTERN "10111000"

:TRIGger{[:ATRigger]][:BTRigger]:PATTe rn:CHANnel<x>

Function Sets or queries the pattern of the specified channel for the combination (pattern) trigger of multiple inputs.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PATTe rn:CHANnel<x> {DONTcare|HIGH|
LOW}
:TRIGger{[:ATRigger]][:BTRigger]:
PATTe rn:CHANnel<x>?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:PATTERN:
CHANNEL1 DONTCARE
:TRIGGER:ATRIGGER:PATTERN:CHANNEL1?
-> :TRIGGER:ATRIGGER:PATTERN:
CHANNEL1 DONTCARE

:TRIGger{[:ATRigger]][:BTRigger]:PATTe rn:CLOCK?

Function Queries all settings related to the clock signal for the combination (pattern) trigger of multiple inputs.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PATTe rn:CLOCK?

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:PATTERN:CLOCK:SLOPe

Function Sets or queries the clock signal slope for the combination (pattern) trigger of multiple inputs.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PATTERN:CLOCK:SLOPe {BOTH|FALL|RISE}
:TRIGger{[:ATRigger]][:BTRigger]:
PATTERN:CLOCK:SLOPe?

Example :TRIGGER:ATRIGGER:PATTERN:CLOCK:
SLOPE FALL
:TRIGGER:ATRIGGER:PATTERN:CLOCK:
SLOPE?
-> :TRIGGER:ATRIGGER:PATTERN:CLOCK:
SLOPE FALL

Description • An error will occur if
:TRIGger:PATTERN:CLOCK:SOURCE NONE
is specified.
• If :TRIGger:SOURCE:CHANNEL<x>:WIND
ow is set to ON, select from {ENTER|EXIT};
otherwise, select from {FALL|RISE}.

:TRIGger{[:ATRigger]][:BTRigger]:PATTERN:CLOCK:SOURce

Function Sets or queries the clock signal for the combination (pattern) trigger of multiple inputs.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PATTERN:CLOCK:SOURce {<NRF>|BIT<x>|
NONE}
:TRIGger{[:ATRigger]][:BTRigger]:
PATTERN:CLOCK:SOURce?

Example :TRIGGER:ATRIGGER:PATTERN:CLOCK:
SOURCE 1
:TRIGGER:ATRIGGER:PATTERN:CLOCK:
SOURCE?
-> :TRIGGER:ATRIGGER:PATTERN:CLOCK:
SOURCE 1

:TRIGger{[:ATRigger]][:BTRigger]:PATTERN:CONDition

Function Sets or queries the trigger condition for the combination (pattern) trigger of multiple inputs.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PATTERN:CONDition {ENTER|EXIT|FALSE|
TRUE}
:TRIGger{[:ATRigger]][:BTRigger]:
PATTERN:CONDition?

Example :TRIGGER:ATRIGGER:PATTERN:
CONDITION ENTER
:TRIGGER:ATRIGGER:PATTERN:CONDITION?
-> :TRIGGER:ATRIGGER:PATTERN:
CONDITION ENTER

:TRIGger{[:ATRigger]][:BTRigger]:PATTERN:LOGic

Function Sets or queries the trigger combination condition for the combination (pattern) trigger of multiple inputs.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PATTERN:LOGic {AND|OR}
:TRIGger{[:ATRigger]][:BTRigger]:
PATTERN:LOGic?

Example :TRIGGER:ATRIGGER:PATTERN:LOGIC AND
:TRIGGER:ATRIGGER:PATTERN:LOGIC?
-> :TRIGGER:ATRIGGER:PATTERN:
LOGIC AND

:TRIGger{[:ATRigger]][:BTRigger]:PATTERN:TQQualify

Function Sets or queries the time condition for combination (pattern) triggering of multiple inputs.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PATTERN:TQQualify {INSide|LTHan|
MTHan|OUTSide|TIMEout}
:TRIGger{[:ATRigger]][:BTRigger]:
PATTERN:TQQualify?

Example :TRIGGER:ATRIGGER:PATTERN:
TQUALIFY INSIDE
:TRIGGER:ATRIGGER:PATTERN:TQUALIFY?
-> :TRIGGER:ATRIGGER:PATTERN:
TQUALIFY INSIDE

:TRIGger{[:ATRigger]][:BTRigger]:PSI5?

Function Queries all PSI5 signal trigger settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5?

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:DATA?

Function Queries all PSI5 signal data trigger settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA?

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:DATA:CONDition

Function Sets or queries the comparison condition of the PSI5 signal data trigger.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:CONDition {EQUal|FALSe|
GREater|INSide|LESS|NOTEqual|
OUTSide|TRUE}
:TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:CONDition?

Example :TRIGGER:ATRIGGER:PSI5:DATA:
CONDITION EQUAL
:TRIGGER:ATRIGGER:PSI5:DATA:
CONDITION?
-> :TRIGGER:ATRIGGER:PSI5:DATA:
CONDITION EQUAL

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:DATA:DECimal<x>

Function Sets or queries the data of the PSI5 signal data trigger in decimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:DECimal<x> {<NRf>}
:TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:DECimal<x>?

<x> = 1 to 2

When <NRf> = 10 bit data, -512 to 511

When <NRf> = 16 bit data, -32768 to 32767

Example :TRIGGER:ATRIGGER:PSI5:DATA:
DECIMAL1 -10
:TRIGGER:ATRIGGER:PSI5:DATA:
DECIMAL1?
-> :TRIGGER:ATRIGGER:PSI5:DATA:
DECIMAL1 -10

Description This command is valid when :TRIGger[:ATRigger|BTRigger]:PSI5:DATA:CONDition is EQUal, GREater, INSide, LESS, NOTEqual, or OUTSide.

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:DATA:DPATtern?

Function Queries all data settings of the PSI5 signal data triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:DPATtern?

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:DATA:DPATtern:HEXa

Function Sets the data of the PSI5 signal data trigger in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:DPATtern:HEXa {<String>}

Example :TRIGGER:ATRIGGER:PSI5:DATA:
DPATTERN:HEXa "3A0F"

Description This command is valid when :TRIGger[:ATRigger|BTRigger]:PSI5:DATA:CONDition is TRUE or FALSe.

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:DATA:DPATtern:PATtern

Function Sets or queries the data of the PSI5 signal data trigger in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:DPATtern:PATtern {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:DPATtern:PATtern?

Example :TRIGGER:ATRIGGER:PSI5:DATA:
DPATTERN:PATTERN "0011010100001111"
:TRIGGER:ATRIGGER:PSI5:DATA:
DPATTERN:PATTERN?
-> :TRIGGER:ATRIGGER:PSI5:DATA:
DPATTERN:PATTERN "0011010100001111"

Description This command is valid when :TRIGger[:ATRigger|BTRigger]:PSI5:DATA:CONDition is TRUE or FALSe.

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:DATA:FRAMEinslot?

Function Queries all slot specification settings of the PSI5 signal data trigger.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:FRAMEinslot?

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:DATA:FRAMEinslot:MODE

Function Sets or queries the slot specification condition (enabled/disabled) of the PSI5 signal data trigger.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:FRAMEinslot:
MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:FRAMEinslot:MODE?

Example :TRIGGER:ATRIGGER:PSI5:DATA:
FRAMEINSLOT:MODE ON
:TRIGGER:ATRIGGER:PSI5:DATA:
FRAMEINSLOT:MODE?
-> :TRIGGER:ATRIGGER:PSI5:DATA:
FRAMEINSLOT:MODE 1

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:DATA:FRAMEinslot:SNUMBER

Function Sets or queries the slot number of the PSI5 signal data trigger.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:FRAMEinslot:
SNUMBER {<NRf>}
:TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:FRAMEinslot:SNUMBER?
<NRf> = 1 to 6

Example :TRIGGER:ATRIGGER:PSI5:DATA:
FRAMEINSLOT:SNUMBER 1
:TRIGGER:ATRIGGER:PSI5:DATA:
FRAMEINSLOT:SNUMBER?
-> :TRIGGER:ATRIGGER:PSI5:DATA:
FRAMEINSLOT:SNUMBER 1

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:DATA:PFORMAT

Function Sets or queries the data input format of the PSI5 signal data trigger in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:PFORMAT {DECimal|HEXa}
:TRIGger{[:ATRigger]][:BTRigger]:
PSI5:DATA:PFORMAT?

Example :TRIGGER:ATRIGGER:PSI5:DATA:
PFORMAT HEXA
:TRIGGER:ATRIGGER:PSI5:DATA:PFORMAT?
-> :TRIGGER:ATRIGGER:PSI5:DATA:
PFORMAT HEXA

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:ERROR?

Function Queries all PSI5 signal error trigger settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:ERROR?

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:ERROR:FNUMBER

Function Sets or queries the PSI5 signal frame number error trigger.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:ERROR:FNUMBER {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
PSI5:ERROR:FNUMBER?

Example :TRIGGER:ATRIGGER:PSI5:ERROR:
FNUMBER ON
:TRIGGER:ATRIGGER:PSI5:ERROR:
FNUMBER?
-> :TRIGGER:ATRIGGER:PSI5:ERROR:
FNUMBER 1

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:ERROR:FRAME

Function Sets or queries the PSI5 signal frame error trigger.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:ERROR:FRAME {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
PSI5:ERROR:FRAME?

Example :TRIGGER:ATRIGGER:PSI5:ERROR:
FRAME ON
:TRIGGER:ATRIGGER:PSI5:ERROR:FRAME?
-> :TRIGGER:ATRIGGER:PSI5:ERROR:
FRAME 1

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:ERROR:MANCHESTER

Function Sets or queries the PSI5 signal Manchester Error trigger.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:ERROR:MANCHESTER {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
PSI5:ERROR:MANCHESTER?

Example :TRIGGER:ATRIGGER:PSI5:ERROR:
MANCHESTER ON
:TRIGGER:ATRIGGER:PSI5:ERROR:
MANCHESTER?
-> :TRIGGER:ATRIGGER:PSI5:ERROR:
MANCHESTER 1

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:ERROR:PCRC

Function Sets or queries the PSI5 signal parity/CRC error trigger.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:ERROR:PCRC {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
PSI5:ERROR:PCRC?

Example :TRIGGER:ATRIGGER:PSI5:ERROR:PCRC ON
:TRIGGER:ATRIGGER:PSI5:ERROR:PCRC?
-> :TRIGGER:ATRIGGER:PSI5:ERROR:
PCRC 1

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:ERROR:SBIT

Function Sets or queries the PSI5 signal start bit error trigger.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:ERROR:SBIT {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
PSI5:ERROR:SBIT?

Example :TRIGGER:ATRIGGER:PSI5:ERROR:SBIT ON
:TRIGGER:ATRIGGER:PSI5:ERROR:SBIT?
-> :TRIGGER:ATRIGGER:PSI5:ERROR:
SBIT 1

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:ERor:SBOundary

Function Sets or queries the PSI5 signal slot boundary error trigger.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:ERor:SBOundary {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:ERor:SBOundary?

Example :TRIGGER:ATRIGGER:PSI5:ERROR:
SBOundary ON
:TRIGGER:ATRIGGER:PSI5:ERROR:
SBOundary?
-> :TRIGGER:ATRIGGER:PSI5:ERROR:
SBOundary 1

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:FRAMeinslot?

Function Queries all PSI5 signal slot specification trigger settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:FRAMeinslot?

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:FRAMeinslot:SNUMber

Function Sets or queries the slot number of the PSI5 signal slot specification trigger.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:FRAMeinslot:SNUMber {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:FRAMeinslot:SNUMber?
<Nrf> = 1 to 6

Example :TRIGGER:ATRIGGER:PSI5:FRAMEINSLOT:
SNUMber 1
:TRIGGER:ATRIGGER:PSI5:FRAMEINSLOT:
SNUMber?
-> :TRIGGER:ATRIGGER:PSI5:
FRAMEINSLOT:SNUMber 1

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:MODE

Function Sets or queries the PSI5 signal trigger mode.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:MODE {DATA|ERROR|FRAMEinslot|
SBIT|SYNC}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:MODE?

Example :TRIGGER:ATRIGGER:PSI5:MODE DATA
:TRIGGER:ATRIGGER:PSI5:MODE?
-> :TRIGGER:ATRIGGER:PSI5:MODE DATA

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup?

Function Queries all bus setup settings for PSI5 signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup?

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA?

Function Queries all PSI5 signal trigger data signal settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA?

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA:BRATe

Function Sets or queries the PSI5 signal trigger bit rate (data transfer rate).

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:BRATe {<Nrf>|USER,
<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:BRATe?
<Nrf> = 125000, 189000
USER <Nrf> = 10000 to 1000000

Example :TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
BRATe 125000
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
BRATe?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
DATA:BRATe 125000

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA:CTOLerance

Function Sets or queries the PSI5 signal trigger clock tolerance range.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:CTOLerance {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:CTOLerance?
<Nrf> = $\pm 0.5\%$ to $\pm 33.3\%$, resolution: 0.1%

Example :TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
CTOLerance 5
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
CTOLerance?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
DATA:CTOLerance 5.00E+00

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA:DBITs

Function Sets or queries the data length for PSI5 signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:DBITs <NRf>
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:DBITs?
<NRf> = 10, 16

Example :TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
DBITS 10
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
DBITS?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
DATA:DBITS 10

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA:EDETectiOn

Function Sets or queries the error detection method for PSI5 signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:EDETectiOn {CRC|
PARity}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:EDETectiOn?

Example :TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
EDETectiOn CRC
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
EDETectiOn?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
DATA:EDETectiOn CRC

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA:SNRejection?

Function Queries all PSI5 signal trigger sync noise rejection settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:SNRejection?

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA:SNRejection:ETIME

Function Sets or queries the rejection end time for PSI5 signal trigger sync noise rejection.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:SNRejection:
ETIME {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:SNRejection:ETIME?
<NRf> = 0 to 0.02 s, resolution: 0.1 us

Example :TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
SNREJECTION:ETIME 66.0000E-06
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
SNREJECTION:ETIME?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
DATA:SNREJECTION:ETIME 66.0000E-06

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA:SNRejection:MODE

Function Sets or queries the on/off state of PSI5 signal trigger sync noise rejection.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:SNRejection:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:SNRejection:MODE?

Example :TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
SNREJECTION:MODE ON
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
SNREJECTION:MODE?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
DATA:SNREJECTION:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:PSI5:SETup:DATA:SOURce

Function Sets or queries the data source for PSI5 signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:SOURce {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:
PSI5:SETup:DATA:SOURce?
<NRf> = 1 to 4

Example :TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
SOURCE 1
:TRIGGER:ATRIGGER:PSI5:SETUP:DATA:
SOURCE?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
DATA:SOURCE 1

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:SETup:NUMBERofslot

Function Sets or queries the number of slots for PSI5 signal trigger.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:SETup:NUMBERofslot {AUTO|<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:
PSI5:SETup:NUMBERofslot?
<Nrf> = 1 to 6

Example :TRIGGER:ATRIGGER:PSI5:SETUP:
NUMBEROFSLOT AUTO
:TRIGGER:ATRIGGER:PSI5:SETUP:
NUMBEROFSLOT?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
NUMBEROFSLOT AUTO

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:SETup:SLOT<x>

Function Sets or queries the slot's start and end times for PSI5 signal trigger.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:SETup:SLOT<x> {<Nrf>[,<Nrf>]}
:TRIGger{[:ATRigger]][:BTRigger]:
PSI5:SETup:SLOT<x>?
<x> = 1 to 6 (slot number)
<Nrf> = 0 to 0.02 s, resolution: 0.1 us

Example :TRIGGER:ATRIGGER:PSI5:SETUP:
SLOT1 44.0000E-06,181.300E-06
:TRIGGER:ATRIGGER:PSI5:SETUP:SLOT1?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
SLOT1 44.0000E-06,181.300E-06

Description The first <Nrf> is the slot's start time and the second the end time. If :TRIGger[:ATRigger]:PSI5:SETup:NUMBERofslot is a number from 1 to 6, when the last slot number is specified with <x>, the second <Nrf> becomes valid. If :TRIGger[:ATRigger]:PSI5:SETupNUMBERofslot is AUTO, query is not possible.

:TRIGger{[:ATRigger]][:BTRigger]:PSI5:SETup:SYNC

Function Sets or queries the sync signal for PSI5 signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
PSI5:SETup:SYNC {<Nrf>|NONE}
:TRIGger{[:ATRigger]][:BTRigger]:
PSI5:SETup:SYNC?
<Nrf> = 1 to 4

Example :TRIGGER:ATRIGGER:PSI5:SETUP:SYNC 1
:TRIGGER:ATRIGGER:PSI5:SETUP:SYNC?
-> :TRIGGER:ATRIGGER:PSI5:SETUP:
SYNC 1

:TRIGger{[:ATRigger]][:BTRigger]:SENT?

Function Queries all SENT signal trigger settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT?

:TRIGger{[:ATRigger]][:BTRigger]:SENT:CEFactor?

Function Queries all SENT signal trigger error factor settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:CEFactor?

:TRIGger{[:ATRigger]][:BTRigger]:SENT:CEFactor:SAComm?

Function Queries all status and communication error factor settings of SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:CEFactor:SAComm?

:TRIGger{[:ATRigger]][:BTRigger]:SENT:CEFactor:SAComm:BIT<x>

Function Sets or queries status and communication error factor bit of SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:CEFactor:SAComm:
BIT<x> {<Boolean>}

:TRIGger{[:ATRigger]][:BTRigger]:
SENT:CEFactor:SAComm:BIT<x>?
<x> = 0 to 1

Example :TRIGGER:ATRIGGER:SENT:CEFACTOR:
SACOMM:BIT1 ON
:TRIGGER:ATRIGGER:SENT:CEFACTOR:
SACOMM:BIT1?
-> :TRIGGER:ATRIGGER:SENT:CEFACTOR:
SCPULSES 1

:TRIGger{[:ATRigger]][:BTRigger]:SENT:CEFactor:SCPulses

Function Sets or queries the consecutive calibration pulse error factor of SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:CEFactor:SCPulses {OFF|OPT2|
POPTION}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:CEFactor:SCPulses?

Example :TRIGGER:ATRIGGER:SENT:CEFACTOR:
SCPULSES OFF
:TRIGGER:ATRIGGER:SENT:CEFACTOR:
SCPULSES?
-> :TRIGGER:ATRIGGER:SENT:CEFACTOR:
SCPULSES OFF

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:SENT:CRCType

Function Sets or queries the SENT signal trigger CRC computation type.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:CRCType {LEGacy|RECommended}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:CRCType?

Example :TRIGGER:ATRIGGER:SENT:
CRCTYPE LEGACY
:TRIGGER:ATRIGGER:SENT:CRCTYPE?
-> :TRIGGER:ATRIGGER:SENT:
CRCTYPE LEGACY

:TRIGger{[:ATRigger]][:BTRigger]:SENT:CTICK

Function Sets or queries the SENT signal trigger clock tick value.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:CTICK {<NRF>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:CTICK?

Example <NRF> = 1.00 μ s to 100.00 μ s, resolution: 0.01 μ s
:TRIGGER:ATRIGGER:SENT:
CTICK 0.000001
:TRIGGER:ATRIGGER:SENT:CTICK?
-> :TRIGGER:ATRIGGER:SENT:
CTICK 1.000000E-06

:TRIGger{[:ATRigger]][:BTRigger]:SENT:CTOLerance

Function Sets or queries the SENT signal trigger clock tolerance.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:CTOLerance {<NRF>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:CTOLerance?

Example <NRF> = $\pm 1.0\%$ to $\pm 30.0\%$, resolution: 0.1%
:TRIGGER:ATRIGGER:SENT:
CTOLERANCE 25.0
:TRIGGER:ATRIGGER:SENT:CTOLERANCE?
-> :TRIGGER:ATRIGGER:SENT:
CTOLERANCE 25.0E+00

:TRIGger{[:ATRigger]][:BTRigger]:SENT:DNIBbles

Function Sets or queries the number of data nibbles for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:DNIBbles {<NRF>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:DNIBbles?

Example <NRF> = 1 to 6
:TRIGGER:ATRIGGER:SENT:DNIBBLES 6
:TRIGGER:ATRIGGER:SENT:DNIBBLES?
-> :TRIGGER:ATRIGGER:SENT:DNIBBLES 6

:TRIGger{[:ATRigger]][:BTRigger]:SENT:DTYPE

Function Sets or queries the SENT signal trigger fast channel data format.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:DTYPE {NIBBLE|USER}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:DTYPE?

Example :TRIGGER:ATRIGGER:SENT:DTYPE NIBBLE
:TRIGGER:ATRIGGER:SENT:DTYPE?
-> :TRIGGER:ATRIGGER:SENT:
DTYPE NIBBLE

:TRIGger{[:ATRigger]][:BTRigger]:SENT:ERRORor?

Function Queries all SENT signal trigger error settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:ERRORor?

:TRIGger{[:ATRigger]][:BTRigger]:SENT:ERROR:FCRC

Function Sets or queries the SENT signal trigger fast channel CRC error.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:ERROR:FCRC {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:ERROR:FCRC?

Example :TRIGGER:ATRIGGER:SENT:ERROR:FCRC ON
:TRIGGER:ATRIGGER:SENT:ERROR:FCRC?
-> :TRIGGER:ATRIGGER:SENT:ERROR:
FCRC 1

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:ER
Ror:NDValue**

Function Sets or queries the SENT signal trigger nibble data value error.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:ERror:NDValue {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:ERror:NDValue?

Example :TRIGGER:ATRIGGER:SENT:ERROR:
NDVALUE ON
:TRIGGER:ATRIGGER:SENT:ERROR:
NDVALUE?
-> :TRIGGER:ATRIGGER:SENT:ERROR:
NDVALUE 1

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:ER
Ror:NNUmber**

Function Sets or queries the SENT signal trigger nibble data count error.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:ERror:NNUmber {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:ERror:NNUmber?

Example :TRIGGER:ATRIGGER:SENT:ERROR:
NNUmber ON
:TRIGGER:ATRIGGER:SENT:ERROR:
NNUmber?
-> :TRIGGER:ATRIGGER:SENT:ERROR:
NNUmber 1

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:ER
Ror:SAComm**

Function Sets or queries the SENT signal trigger status and communication error.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:ERror:SAComm {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:ERror:SAComm?

Example :TRIGGER:ATRIGGER:SENT:ERROR:
SAComm ON
:TRIGGER:ATRIGGER:SENT:ERROR:SAComm?
-> :TRIGGER:ATRIGGER:SENT:ERROR:
SAComm 1

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:ER
Ror:SCPulses**

Function Sets or queries the consecutive calibration pulse error for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:ERror:SCPulses {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:ERror:SCPulses?

Example :TRIGGER:ATRIGGER:SENT:ERROR:
SCPULSES ON
:TRIGGER:ATRIGGER:SENT:ERROR:
SCPULSES?
-> :TRIGGER:ATRIGGER:SENT:ERROR:
SCPULSES 1

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:ER
Ror:SCRC**

Function Sets or queries the SENT signal trigger slow channel CRC error.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:ERror:SCRC {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:ERror:SCRC?

Example :TRIGGER:ATRIGGER:SENT:ERROR:SCRC ON
:TRIGGER:ATRIGGER:SENT:ERROR:SCRC?
-> :TRIGGER:ATRIGGER:SENT:ERROR:
SCRC 1

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:FD
ATa?**

Function Queries all fast channel data settings for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:FDATa?

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:FD
ATa:DATA<x>?**

Function Queries all settings related to the specified fast channel user data for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:FDATa:DATA<x>?
<x> = 1 to 4

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:SENT:FDATa:DATA<x>:CONDition

Function Sets or queries the comparison condition of the specified user data of the SENT signal trigger fast channel.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:FDATa:DATA<x>:CONDition {EQUal|
GREater|INSide|LESS|NOTEqual|
OUTSide}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:FDATa:DATA<x>:CONDition?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:SENT:FDATA:DATA1:
CONDITION EQUAL
:TRIGGER:ATRIGGER:SENT:FDATA:DATA1:
CONDITION?
-> :TRIGGER:ATRIGGER:SENT:FDATA:
DATA1:CONDITION EQUAL

:TRIGger{[:ATRigger]][:BTRigger]:SENT:FDATa:DATA<x>:DECimal<y>

Function Sets or queries the data of the specified user data of the SENT signal trigger fast channel in decimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:FDATa:DATA<x>:
DECimal<y> {<NRF>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:FDATa:DATA<x>:DECimal<y>?
<x> = 1 to 4
<y> = 1 to 2
<NRF> = See the Features Guide for this
information.

Example :TRIGGER:ATRIGGER:SENT:FDATA:DATA1:
DECIMAL1 1
:TRIGGER:ATRIGGER:SENT:FDATA:DATA1:
DECIMAL1?
-> :TRIGGER:ATRIGGER:SENT:FDATA:
DATA1:DECIMAL1 1

:TRIGger{[:ATRigger]][:BTRigger]:SENT:FDATa:DATA<x>:MODE

Function Sets or queries the data enable/disable condition of the specified user data of the SENT signal trigger fast channel.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:FDATa:DATA<x>:MODE{<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:FDATa:DATA<x>:MODE?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:SENT:FDATA:DATA1:
MODE ON
:TRIGGER:ATRIGGER:SENT:FDATA:DATA1:
MODE?
-> :TRIGGER:ATRIGGER:SENT:FDATA:
DATA1:MODE 1

:TRIGger{[:ATRigger]][:BTRigger]:SENT:FDATa:DNIBbles?

Function Queries all fast channel nibble data settings for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:FDATa:DNIBbles?

:TRIGger{[:ATRigger]][:BTRigger]:SENT:FDATa:DNIBbles:CONDition

Function Sets or queries the nibble data comparison condition of the SENT signal trigger fast channel.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:FDATa:DNIBbles:
CONDition {FALSE|TRUE}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:FDATa:DNIBbles:CONDition?

Example :TRIGGER:ATRIGGER:SENT:FDATA:
DNIBBLES:CONDITION TRUE
:TRIGGER:ATRIGGER:SENT:FDATA:
DNIBBLES:CONDITION?
-> :TRIGGER:ATRIGGER:SENT:FDATA:
DNIBBLES:CONDITION TRUE

:TRIGger{[:ATRigger]][:BTRigger]:SENT:FDATa:DNIBbles:HEXa

Function Sets the SENT signal trigger fast channel nibble data in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:FDATa:DNIBbles:HEXa {<String>}

Example :TRIGGER:ATRIGGER:SENT:FDATA:
DNIBBLES:HEXa "112233"

:TRIGger{[:ATRigger]][:BTRigger]:SENT:FDATa:DNIBbles:PATtern

Function Sets or queries the SENT signal trigger fast channel nibble data in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:FDATa:DNIBbles:
PATtern {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:FDATa:DNIBbles:PATtern?

Example :TRIGGER:ATRIGGER:SENT:FDATA:
DNIBBLES:
PATTERN "111100001111000011110000"
:TRIGGER:ATRIGGER:SENT:FDATA:
DNIBBLES:PATTERN?
-> :TRIGGER:ATRIGGER:SENT:FDATA:
DNIBBLES:
PATTERN "111100001111000011110000"

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:FS
AComm?**

Function Queries all fast channel status and communication nibble settings for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:FSAComm?

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:FS
AComm:HEXa**

Function Sets the SENT signal trigger fast channel status and communication nibble data in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:FSAComm:HEXa {<String>}

Example :TRIGGER:ATRIGGER:SENT:FSACOMM:
HEXa "F"

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:FS
AComm:PATtern**

Function Sets or queries the SENT signal trigger fast channel status and communication nibble data in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:FSAComm:PATtern {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:FSAComm:PATtern?

Example :TRIGGER:ATRIGGER:SENT:FSACOMM:
PATTERN "1111"
:TRIGGER:ATRIGGER:SENT:FSACOMM:
PATTERN?
-> :TRIGGER:ATRIGGER:SENT:FSACOMM:
PATTERN "1111"

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:MO
DE**

Function Sets or queries the SENT signal trigger mode.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:MODE {EFAST|ERROR|ESLow|FData|
FSAComm|SDaTa}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:MODE?

Example :TRIGGER:ATRIGGER:SENT:MODE ERROR
:TRIGGER:ATRIGGER:SENT:MODE?
-> :TRIGGER:ATRIGGER:SENT:MODE ERROR

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:PP
ULse**

Function Sets or queries the presence or absence of pause pulses for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:PPULse {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:PPULse?

Example :TRIGGER:ATRIGGER:SENT:PPULSE ON
:TRIGger:ATRIGGER:SENT:PPULSE?
-> :TRIGGER:ATRIGGER:SENT:PPULSE 1

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa?**

Function Queries all slow channel data settings for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa?

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:ENHanced?**

Function Queries all slow channel enhanced type settings for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced?

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:ENHanced:CBIT**

Function Sets or queries the SENT signal trigger slow channel enhanced type configuration bit.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:CBIT {D12Bit|
D16Bit}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:CBIT?

Example :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:CBIT D12Bit
:TRIGger:ATRIGGER:SENT:SDaTa:
ENHANCED:CBIT?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:CBIT D12Bit

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:ENHanced:D12Bit?**

Function Queries all slow channel enhanced type 12-bit data and 8-bit ID settings for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D12Bit?

5.33 TRIGger Group

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:ENHanced:D12Bit:DATA?**

Function Queries all slow channel enhanced type 12-bit data and 8-bit ID data settings for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D12Bit:DATA?

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:ENHanced:D12Bit:DATA:CONDition**

Function Sets or queries the data comparison condition of slow channel enhanced type 12-bit data and 8-bit ID for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D12Bit:DATA:
CONDition {EQUAL|FALSE|GREATER|
INSide|LESS|NOTEQUAL|OUTSide|TRUE}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D12Bit:DATA:
CONDition?

Example :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:CONDITION EQUAL
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:CONDITION?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:CONDITION EQUAL

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:ENHanced:D12Bit:DATA:DECimal<x>**

Function Sets or queries the slow channel enhanced type 12-bit data and 8-bit ID data for SENT signal triggering in decimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D12Bit:DATA:
DECimal<x> {<NRF>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D12Bit:DATA:
DECimal<x>?
<x> = 1 to 2
<NRF> = See the Features Guide for this
information.

Example :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:DECIMAL1 123
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:DECIMAL1?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:DECIMAL1 123

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:ENHanced:D12Bit:DATA:HEXa**

Function Sets the slow channel enhanced type 12-bit data and 8-bit ID data for SENT signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D12Bit:DATA:
HEXa {<String>}

Example :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:HEXA "123"

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:ENHanced:D12Bit:DATA:MODE**

Function Sets or queries the data enable/disable condition of the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D12Bit:DATA:
MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D12Bit:DATA:
MODE?

Example :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:MODE ON
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:MODE?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:MODE 1

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:ENHanced:D12Bit:DATA:PATtern**

Function Sets the slow channel enhanced type 12-bit data and 8-bit ID data for SENT signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D12Bit:DATA:
PATtern {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D12Bit:DATA:
PATtern?

Example :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:
PATTERN "111101010101"
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:PATTERN?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D12BIT:DATA:
PATTERN "111101010101"

**:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD
ATa:ENHanced:D12Bit:ID?**

Function Queries all ID settings related to the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D12Bit:ID?

**:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD
ATa:ENHanced:D12Bit:ID:CONDition**

Function Sets or queries the ID comparison condition of slow channel enhanced type 12-bit data and 8-bit ID for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D12Bit:ID:
CONDition {EQual|GREater|INSide|
LESS|NOTEqual|OUTSide}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D12Bit:ID:
CONDition?

Example :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D12BIT:ID:CONDITION EQUAL
:TRIGger:ATRIGGER:SENT:SDaTa:
ENHANCED:D12BIT:ID:CONDITION?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D12BIT:ID:CONDITION EQUAL

**:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD
ATa:ENHanced:D12Bit:ID:DECimal<x>**

Function Sets or queries the ID of the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal triggering in decimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D12Bit:ID:
DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D12Bit:ID:
DECimal<x>?
<x> = 1 to 2

Example :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D12BIT:ID:DECIMAL1 0
:TRIGger:ATRIGGER:SENT:SDaTa:
ENHANCED:D12BIT:ID:DECIMAL1?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D12BIT:ID:DECIMAL1 0

**:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD
ATa:ENHanced:D12Bit:ID:MODE**

Function Sets or queries the ID enable/disable condition of the slow channel enhanced type 12-bit data and 8-bit ID for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D12Bit:ID:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D12Bit:ID:MODE?

Example :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D12BIT:ID:MODE ON
:TRIGger:ATRIGGER:SENT:SDaTa:
ENHANCED:D12BIT:ID:MODE?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D12BIT:ID:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD
ATa:ENHanced:D12Bit:PFORMAT**

Function Sets the data input format, which is one of the slow channel enhanced type 12-bit data and 8-bit ID data conditions, for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D12Bit:
PFORMAT {DECimal|HEXa}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D12Bit:PFORMAT?

Example :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D12BIT:PFORMAT HEXA
:TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D12BIT:PFORMAT?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D12BIT:PFORMAT HEXA

**:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD
ATa:ENHanced:D16Bit?**

Function Queries all slow channel enhanced type 16-bit data and 4-bit ID settings for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit?

**:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD
ATa:ENHanced:D16Bit:DATA?**

Function Queries all slow channel enhanced type 16-bit data and 4-bit ID data settings for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA?

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD ATa:ENHanced:D16Bit:DATA:CONDition

Function Sets or queries the data comparison condition of slow channel enhanced type 16-bit data and 4-bit ID for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
CONDition {EQUAL|FALSE|GREater|
INSide|LESS|NOTEqual|OUTSide|TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
CONDition?

Example :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:CONDITION EQUAL
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:CONDITION?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:CONDITION EQUAL

:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD ATa:ENHanced:D16Bit:DATA:DECimal<x>

Function Sets or queries the slow channel enhanced type 16-bit data and 4-bit ID data for SENT signal triggering in decimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
DECimal<x>?

<x> = 1 to 2

<Nrf> = See the Features Guide for this information.

Example :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:DECIMAL 123
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:DECIMAL?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:DECIMAL 123

:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD ATa:ENHanced:D16Bit:DATA:HEXa

Function Sets the slow channel enhanced type 16-bit data and 4-bit ID data for SENT signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
HEXa {<String>}

Example :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:HEXA "0000"

:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD ATa:ENHanced:D16Bit:DATA:MODE

Function Sets or queries the data enable/disable condition of the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
MODE?

Example :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:MODE ON
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:MODE?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD ATa:ENHanced:D16Bit:DATA:PATtern

Function Sets or queries the slow channel enhanced type 16-bit data and 4-bit ID data for SENT signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
PATtern {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:DATA:
PATtern?

Example :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:
PATTERN "1111000011110000"
:TRIGger:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:PATTERN?
-> :TRIGGER:ATRIGGER:SENT:SDATA:
ENHANCED:D16BIT:DATA:
PATTERN "1111000011110000"

:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD ATa:ENHanced:D16Bit:ID?

Function Queries all ID settings related to the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:ENHanced:D16Bit:ID?

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:ENHanced:D16Bit:ID:CONDition**

Function Sets or queries the ID comparison condition of slow channel enhanced type 16-bit data and 4-bit ID for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D16Bit:ID:
CONDition {EQUal|GREater|INSide|
LESS|NOTEqual|OUTSide}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D16Bit:ID:
CONDition?

Example :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D16BIT:ID:CONDITION EQUAL
:TRIGger:ATRIGGER:SENT:SDaTa:
ENHANCED:D16BIT:ID:CONDITION?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D16BIT:ID:CONDITION EQUAL

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:ENHanced:D16Bit:ID:DECimal<x>**

Function Sets or queries the ID of the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal triggering in decimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D16Bit:ID:
DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D16Bit:ID:
DECimal<x>?
<x> = 1 to 2

Example :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D16BIT:ID:DECIMAL1 0
:TRIGger:ATRIGGER:SENT:SDaTa:
ENHANCED:D16BIT:ID:DECIMAL1?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D16BIT:ID:DECIMAL1 0

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:ENHanced:D16Bit:ID:MODE**

Function Sets or queries the ID enable/disable condition of the slow channel enhanced type 16-bit data and 4-bit ID for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D16Bit:ID:
MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D16Bit:ID:MODE?

Example :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D16BIT:ID:MODE ON
:TRIGger:ATRIGGER:SENT:SDaTa:
ENHANCED:D16BIT:ID:MODE?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D16BIT:ID:MODE 1

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:ENHanced:D16Bit:PFORMAT**

Function Sets the data input format, which is one of the slow channel enhanced type 16-bit data and 4-bit ID data conditions, for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D16Bit:
PFORMAT {DECimal|HEXa}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:ENHanced:D16Bit:PFORMAT?

Example :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D16BIT:PFORMAT HEXA
:TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D16BIT:PFORMAT?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
ENHANCED:D16BIT:PFORMAT HEXA

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:SHORT?**

Function Queries all slow channel short type settings for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:SHORT?

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:SHORT:DATA?**

Function Queries all slow channel short type data settings for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:SHORT:DATA?

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:SHORT:DATA:CONDition**

Function Sets or queries the short type data comparison condition of the SENT signal trigger slow channel.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:SHORT:DATA:
CONDition {EQUal|FALSE|GREater|
INSide|LESS|NOTEqual|OUTSide|TRUE}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:SHORT:DATA:CONDition?

Example :TRIGGER:ATRIGGER:SENT:SDaTa:SHORT:
DATA:CONDITION EQUAL
:TRIGger:ATRIGGER:SENT:SDaTa:SHORT:
DATA:CONDITION?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
SHORT:DATA:CONDITION EQUAL

5.33 TRIGger Group

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:SHORT:DATA:DECimal<x>**

Function Sets or queries the SENT signal trigger slow channel short type data in decimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:SHORT:DATA:
DECimal<x> {<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:SHORT:DATA:DECimal<x>?
<x> = 1 to 2
<Nrf> = See the Features Guide for this
information.

Example :TRIGGER:ATRIGGER:SENT:SDaTa:SHORT:
DATA:DECIMAL1 123
:TRIGGER:ATRIGGER:SENT:SDaTa:SHORT:
DATA:DECIMAL1?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
SHORT:DATA:DECIMAL1 123

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:SHORT:DATA:HEXa**

Function Sets the SENT signal trigger slow channel short type data in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:SHORT:DATA:HEXa {<String>}

Example :TRIGGER:ATRIGGER:SENT:SDaTa:SHORT:
DATA:HEXa "00"

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:SHORT:DATA:MODE**

Function Sets or queries the short type data enable/disable condition of the SENT signal trigger slow channel.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:SHORT:DATA:
MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:SHORT:DATA:MODE?

Example :TRIGGER:ATRIGGER:SENT:SDaTa:SHORT:
DATA:MODE ON
:TRIGGER:ATRIGGER:SENT:SDaTa:SHORT:
DATA:MODE?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
SHORT:DATA:MODE 1

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:SHORT:DATA:PATtern**

Function Sets or queries the SENT signal trigger slow channel short type data in binary notation.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:SHORT:DATA:
PATtern {<String>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:SHORT:DATA:PATtern?

Example :TRIGGER:ATRIGGER:SENT:SDaTa:SHORT:
DATA:PATTERN "11110000"
:TRIGGER:ATRIGGER:SENT:SDaTa:SHORT:
DATA:PATTERN?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
SHORT:DATA:PATTERN "11110000"

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:SHORT:ID?**

Function Queries all slow channel short type ID settings for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:SHORT:ID?

**:TRIGger{[:ATRigger]][:BTRigger]:SENT:SD
ATa:SHORT:ID:CONDition**

Function Sets or queries the short type ID comparison condition of the SENT signal trigger slow channel.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:SHORT:ID:
CONDition {EQUAL|GREATER|INSide|
LESS|NOTEQUAL|OUTSide}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:SDaTa:SHORT:ID:CONDition?

Example :TRIGGER:ATRIGGER:SENT:SDaTa:SHORT:
ID:CONDITION EQUAL
:TRIGGER:ATRIGGER:SENT:SDaTa:SHORT:
ID:CONDITION?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
SHORT:ID:CONDITION EQUAL

**:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD
ATa:SHORT:ID:DECimal<x>**

Function Sets or queries the SENT signal trigger slow channel short type ID in decimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORT:ID:
DECimal<x> {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORT:ID:DECimal<x>?
<x> = 1 to 2

Example :TRIGGER:ATRIGGER:SENT:SDaTa:SHORT:
ID:DECimal1 0
:TRIGger:ATRIGGER:SENT:SDaTa:SHORT:
ID:DECIMAL1?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
SHORT:ID:DECIMAL1 0

**:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD
ATa:SHORT:ID:MODE**

Function Sets or queries the short type ID enable/disable condition of the SENT signal trigger slow channel.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORT:ID:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORT:ID:MODE?

Example :TRIGGER:ATRIGGER:SENT:SDaTa:SHORT:
ID:MODE ON
:TRIGger:ATRIGGER:SENT:SDaTa:SHORT:
ID:MODE?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
SHORT:ID:MODE 1

**:TRIGger{[:ATRigger]|:BTRigger}:SENT:SD
ATa:SHORT:PFormat**

Function Sets or queries the data input format, which is one of the slow channel short type data conditions, for SENT signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORT:PFormat {DECimal|
HEXa}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SDaTa:SHORT:PFormat?

Example :TRIGGER:ATRIGGER:SENT:SDaTa:SHORT:
PFORMAT HEXA
:TRIGger:ATRIGGER:SENT:SDaTa:SHORT:
PFORMAT?
-> :TRIGGER:ATRIGGER:SENT:SDaTa:
SHORT:PFORMAT HEXA

**:TRIGger{[:ATRigger]|:BTRigger}:SENT:SO
URce**

Function Sets or queries the SENT signal trigger source.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:SOURce {<NRf>|BIT<x>}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:SOURce?
<NRf> = 1 to 4
BIT<x>'s <x> = 1 to 8

Example :TRIGGER:ATRIGGER:SENT:SOURCE 1
:TRIGger:ATRIGGER:SENT:SOURCE?
-> :TRIGGER:ATRIGGER:SENT:SOURCE 1

**:TRIGger{[:ATRigger]|:BTRigger}:SENT:ST
YPe**

Function Sets or queries the SENT signal trigger slow channel format.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:STYPe {ENHanced|SHORT}
:TRIGger{[:ATRigger]|:BTRigger}:
SENT:STYPe?

Example :TRIGGER:ATRIGGER:SENT:
STYPe ENHANCED
:TRIGger:ATRIGGER:SENT:STYPe?
-> :TRIGGER:ATRIGGER:SENT:
STYPe ENHANCED

**:TRIGger{[:ATRigger]|:BTRigger}:SENT:US
ETup?**

Function Queries all settings related to user-defined data of the SENT signal trigger fast channel.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:USETup?

**:TRIGger{[:ATRigger]|:BTRigger}:SENT:US
ETup:DATA<x>?**

Function Queries all settings related to user-defined data of the SENT signal trigger fast channel.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SENT:USETup:DATA<x>?
<x> = 1 to 4

5.33 TRIGger Group

:TRIGger{[:ATRigger]][:BTRigger]:SENT:US ETup:DATA<x>:MODE

Function Sets or queries the enable/disable condition of the specified user-defined data of the SENT signal trigger fast channel.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:USETup:DATA<x>:MODE {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:USETup:DATA<x>:MODE?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:SENT:USETUP:DATA1:
MODE ON
:TRIGger:ATRIGGER:SENT:USETUP:DATA1:
MODE?
-> :TRIGGER:ATRIGGER:SENT:USETUP:
DATA1:MODE 1

:TRIGger{[:ATRigger]][:BTRigger]:SENT:US ETup:DATA<x>:ORDER

Function Sets or queries the byte order of the specified user-defined data of the SENT signal trigger fast channel.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:USETup:DATA<x>:ORDER {BIG|
LITTLE}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:USETup:DATA<x>:ORDER?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:SENT:USETUP:DATA1:
ORDER BIG
:TRIGger:ATRIGGER:SENT:USETUP:DATA1:
ORDER?
-> :TRIGGER:ATRIGGER:SENT:USETUP:
DATA1:ORDER BIG

:TRIGger{[:ATRigger]][:BTRigger]:SENT:US ETup:DATA<x>:SIZE

Function Sets or queries the data size of the specified user-defined data of the SENT signal trigger fast channel.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:USETup:DATA<x>:SIZE {<NRf>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:USETup:DATA<x>:SIZE?
<x> = 1 to 4
<NRf> = 0 to 24

Example :TRIGGER:ATRIGGER:SENT:USETUP:DATA1:
SIZE 24
:TRIGger:ATRIGGER:SENT:USETUP:DATA1:
SIZE?
-> :TRIGGER:ATRIGGER:SENT:USETUP:
DATA1:SIZE 24

:TRIGger{[:ATRigger]][:BTRigger]:SENT:US ETup:MULTiplexing

Function Sets or queries the multiplexing enable/disable condition of the specified user-defined data of the SENT signal trigger fast channel.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:USETup:MULTiplexing {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:USETup:MULTiplexing?

Example :TRIGGER:ATRIGGER:SENT:USETUP:
MULTIPLEXING ON
:TRIGGER:ATRIGGER:SENT:USETUP:
MULTIPLEXING?
-> :TRIGGER:ATRIGGER:SENT:USETUP:
MULTIPLEXING ON

:TRIGger{[:ATRigger]][:BTRigger]:SENT:VE RSion

Function Sets or queries the SENT signal trigger specification version.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SENT:VERSion {APR2016|FEB2008|
JAN2010}
:TRIGger{[:ATRigger]][:BTRigger]:
SENT:VERESion?

Example :TRIGGER:ATRIGGER:SENT:
VERSION JAN2010
:TRIGger:ATRIGGER:SENT:VERSION?
-> :TRIGGER:ATRIGGER:SENT:
VERSION JAN2010

:TRIGger{[:ATRigger]][:BTRigger]:SIMPLE?

Function Queries all edge trigger settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
SIMPLE?

:TRIGger{[:ATRigger]|:BTRigger}:SIMple:HFRejection(HighFrequencyREJECTION)

Function Sets or queries the on/off state of the low-pass filter (HF rejection) for the edge trigger source.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SIMple:HFRejection {<Frequency>|OFF}
:TRIGger{[:ATRigger]|:BTRigger}:SIMple:HFRejection?

Example Frequency: 15 kHz to 20 MHz

```
:TRIGGER:ATRIGGER:SIMPLE:
HFREJECTION 15KHZ
:TRIGGER:ATRIGGER:SIMPLE:
HFREJECTION?
-> :TRIGGER:ATRIGGER:SIMPLE:
HFREJECTION 15000
```

Description You cannot use this command or query if :TRIGger{[:ATRigger]|:BTRigger}:SIMple:SOURce is set to EXTERNAL or LINE.

:TRIGger{[:ATRigger]|:BTRigger}:SIMple:LEVel

Function Sets or queries the trigger level of the edge trigger source.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SIMple:LEVel {<Voltage>}
:TRIGger{[:ATRigger]|:BTRigger}:SIMple:LEVel?

<Voltage> = See the Features Guide for this information.

Example :TRIGGER:ATRIGGER:SIMPLE:LEVEL 0V
:TRIGGER:ATRIGGER:SIMPLE:LEVEL?
-> :TRIGGER:ATRIGGER:SIMPLE:LEVEL 0.000E+00

Description • You cannot use this command or query if :TRIGger{[:ATRigger]|:BTRigger}:SIMple:SOURce is set to LINE.
• When a current probe is in use, this command does not set or query the <Current> value.

:TRIGger{[:ATRigger]|:BTRigger}:SIMple:NREJection

Function Sets or queries the noise rejection of the edge trigger source level.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SIMple:NREJection {HIGH|LOW|MIDDLE}
:TRIGger{[:ATRigger]|:BTRigger}:SIMple:NREJection?

Example :TRIGGER:ATRIGGER:SIMPLE:NREJECTION HIGH
:TRIGGER:ATRIGGER:SIMPLE:NREJECTION?
-> :TRIGGER:ATRIGGER:SIMPLE:NREJECTION HIGH

Description You cannot use this command or query if :TRIGger{[:ATRigger]|:BTRigger}:SIMple:SOURce is set to EXTERNAL or LINE.

:TRIGger{[:ATRigger]|:BTRigger}:SIMple:PROBe

Function Sets or queries the probe of the external edge trigger source.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SIMple:PROBe {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:SIMple:PROBe?
<NRf> = 1, 10

Example :TRIGGER:ATRIGGER:SIMPLE:PROBE 1
:TRIGGER:ATRIGGER:SIMPLE:PROBE?
-> :TRIGGER:ATRIGGER:SIMPLE:PROBE 1

Description You cannot use this command or query if :TRIGger{[:ATRigger]|:BTRigger}:SIMple:SOURce is not set to EXTERNAL.

:TRIGger{[:ATRigger]|:BTRigger}:SIMple:RANGe

Function Sets or queries the input range of the external edge trigger source.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SIMple:RANGe {<Voltage>}
:TRIGger{[:ATRigger]|:BTRigger}:SIMple:RANGe?
<Voltage> = 2 to 200 (V)

:TRIGger{[:ATRigger]|:BTRigger}:SIMple:SLOPe

Function Sets or queries the trigger slope (polarity if windows is ON) of the edge trigger source.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SIMple:SLOPe {BOTH|FALL|RISE}
:TRIGger{[:ATRigger]|:BTRigger}:SIMple:SLOPe?

Example :TRIGGER:ATRIGGER:SIMPLE:SLOPE BOTH
:TRIGGER:ATRIGGER:SIMPLE:SLOPE?
-> :TRIGGER:ATRIGGER:SIMPLE:SLOPE BOTH

Description You cannot use this command or query if :TRIGger{[:ATRigger]|:BTRigger}:SIMple:SOURce is set to LINE.

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:SIMPlE:SOURce

Function Sets or queries the edge trigger source.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SIMPlE:SOURce {<Nrf>|BIT<x>|
EXTErnal|LINE}
:TRIGger{[:ATRigger]|:BTRigger}:
SIMPlE:SOURce?
<Nrf> = 1 to 4
BIT<x>'s <x> = 1 to 8

Example :TRIGGER:ATRIGGER:SIMPLE:SOURCE 1
:TRIGGER:ATRIGGER:SIMPLE:SOURCE?
-> :TRIGGER:ATRIGGER:SIMPLE:SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:SPATte rn?

Function Queries all user-defined bus signal trigger settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern?

:TRIGger{[:ATRigger]|:BTRigger}:SPATte rn:BITSize

Function Sets or queries the user-defined bus signal trigger bit length.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:BITSize {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:BITSize?
<Nrf> = 1 to 128

Example :TRIGGER:ATRIGGER:SPATTERN:BITSIZE 1
:TRIGGER:ATRIGGER:SPATTERN:BITSIZE?
-> :TRIGGER:ATRIGGER:SPATTERN:
BITSIZE 1

:TRIGger{[:ATRigger]|:BTRigger}:SPATte rn:BRATe

Function Sets or queries the user-defined bus signal trigger bit rate.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:BRATe {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:BRATe?
<Nrf> = 1000 to 50000000

Example :TRIGGER:ATRIGGER:SPATTERN:
BRATE 1000
:TRIGGER:ATRIGGER:SPATTERN:BRATE?
-> :TRIGGER:ATRIGGER:SPATTERN:
BRATE 1000

:TRIGger{[:ATRigger]|:BTRigger}:SPATte rn:CLOCK?

Function Queries all user-defined bus signal trigger clock signal settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CLOCK?

:TRIGger{[:ATRigger]|:BTRigger}:SPATte rn:CLOCK:MODE

Function Sets or queries the enable/disable condition of the user-defined bus signal trigger clock signal.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CLOCK:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CLOCK:MODE?

Example :TRIGGER:ATRIGGER:SPATTERN:CLOCK:
MODE ON
:TRIGGER:ATRIGGER:SPATTERN:CLOCK:
MODE?
-> :TRIGGER:ATRIGGER:SPATTERN:CLOCK:
MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:SPATte rn:CLOCK:POLarity

Function Sets or queries the polarity of the user-defined bus signal trigger clock signal.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CLOCK:POLarity {FALL|RISE}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CLOCK:POLarity?

Example :TRIGGER:ATRIGGER:SPATTERN:CLOCK:
POLARITY FALL
:TRIGGER:ATRIGGER:SPATTERN:CLOCK:
POLARITY?
-> :TRIGGER:ATRIGGER:SPATTERN:CLOCK:
POLARITY FALL

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:CLOCK:SOURce

Function Sets or queries the user-defined bus signal trigger clock signal.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CLOCK:SOURce {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CLOCK:SOURce?
<Nrf> = 1 to 4

Example :TRIGGER:ATRIGGER:SPATTERN:CLOCK:
SOURCE 1
:TRIGGER:ATRIGGER:SPATTERN:CLOCK:
SOURCE?
-> :TRIGGER:ATRIGGER:SPATTERN:CLOCK:
SOURCE 1

Description The clock signal that you can specify varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:SPATtern:DATA:SOURce setting. For details, see the Features Guide.

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:CS?

Function Queries all user-defined bus signal trigger chip select signal settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CS?

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:CS:ACTiVe

Function Sets or queries the active state of the user-defined bus signal trigger chip select signal.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CS:ACTiVe {HIGH|LOW}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CS:ACTiVe?

Example :TRIGGER:ATRIGGER:SPATTERN:CS:
ACTIVE HIGH
:TRIGGER:ATRIGGER:SPATTERN:CS:
ACTIVE?
-> :TRIGGER:ATRIGGER:SPATTERN:CS:
ACTIVE HIGH

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:CS:SOURce

Function Sets or queries the user-defined bus signal trigger chip select signal.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CS:SOURce {<Nrf>|NONE}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:CS:SOURce?
<Nrf> = 1 to 4

Example :TRIGGER:ATRIGGER:SPATTERN:CS:
SOURCE 1
:TRIGGER:ATRIGGER:SPATTERN:CS:
SOURCE?
-> :TRIGGER:ATRIGGER:SPATTERN:CS:
SOURCE 1

Description The chip select signal that you can specify varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:SPATtern:DATA:SOURce setting. For details, see the Features Guide.

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:DATA?

Function Queries all user-defined bus signal trigger data signal settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:DATA?

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:DATA:ACTiVe

Function Sets or queries the active state of the user-defined bus signal trigger data signal.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:DATA:ACTiVe {HIGH|LOW}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:DATA:ACTiVe?

Example :TRIGGER:ATRIGGER:SPATTERN:DATA:
ACTIVE HIGH
:TRIGGER:ATRIGGER:SPATTERN:DATA:
ACTIVE?
-> :TRIGGER:ATRIGGER:SPATTERN:DATA:
ACTIVE HIGH

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:DATA:SOURce

Function Sets or queries the user-defined bus signal trigger data signal.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:DATA:SOURce {<NRf>}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:DATA:SOURce?
<NRf> = 1 to 4

Example :TRIGGER:ATRIGGER:SPATTERN:DATA:
SOURCE 1
:TRIGGER:ATRIGGER:SPATTERN:DATA:
SOURCE?
-> :TRIGGER:ATRIGGER:SPATTERN:DATA:
SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:HEXa

Function Queries the data conditions of the user-defined bus signal trigger in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:HEXa {<String>}

Example :TRIGGER:ATRIGGER:SPATTERN:
HEXa "12"

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:LATCh?

Function Queries all user-defined bus signal trigger latch signal settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:LATCh?

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:LATCh:POLarity

Function Sets or queries the polarity of the user-defined bus signal trigger latch signal.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:LATCh:POLarity {FALL|RISE}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:LATCh:POLarity?

Example :TRIGGER:ATRIGGER:SPATTERN:LATCH:
POLARITY FALL
:TRIGGER:ATRIGGER:SPATTERN:LATCH:
POLARITY?
-> :TRIGGER:ATRIGGER:SPATTERN:LATCH:
POLARITY FALL

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:LATCh:SOURce

Function Sets or queries the user-defined bus signal trigger latch signal.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:LATCh:SOURce {<NRf>|NONE}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:LATCh:SOURce?
<NRf> = 1 to 4

Example :TRIGGER:ATRIGGER:SPATTERN:LATCH:
SOURCE 1
:TRIGGER:ATRIGGER:SPATTERN:LATCH:
SOURCE?
-> :TRIGGER:ATRIGGER:SPATTERN:LATCH:
SOURCE 1

Description The latch signal that you can specify varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:SPATtern:DATA:SOURce setting. For details, see the Features Guide.

:TRIGger{[:ATRigger]|:BTRigger}:SPATtern:PATtern

Function Sets or queries the data conditions of the user-defined bus signal trigger in binary notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:PATtern {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:
SPATtern:PATtern?

Example :TRIGGER:ATRIGGER:SPATTERN:
PATTERN "00110101"
:TRIGGER:ATRIGGER:SPATTERN:PATTERN?
-> :TRIGGER:ATRIGGER:SPATTERN:
PATTERN "00110101"

:TRIGger{[:ATRigger]|:BTRigger}:SPI?

Function Queries all SPI bus signal trigger settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SPI?

:TRIGger{[:ATRigger]|:BTRigger}:SPI:BIT order

Function Sets or queries the bit order of the SPI bus signal trigger data.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SPI:
BITorder {LSBFirst|MSBFirst}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:
BITorder?

Example :TRIGGER:ATRIGGER:SPI:
BITORDER LSBFIRST
:TRIGGER:ATRIGGER:SPI:BITORDER?
-> :TRIGGER:ATRIGGER:SPI:
BITORDER LSBFIRST

:TRIGger{[:ATRigger]][:BTRigger]:SPI:CLOCK?

Function Queries all SPI bus signal trigger clock signal settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:SPI:CLOCK?

:TRIGger{[:ATRigger]][:BTRigger]:SPI:CLOCK:POLarity

Function Sets or queries the polarity of the SPI bus signal trigger clock signal.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:SPI:CLOCK:POLarity {FALL|RISE}
:TRIGger{[:ATRigger]][:BTRigger]:SPI:CLOCK:POLarity?

Example :TRIGGER:ATRIGGER:SPI:CLOCK:
POLARITY FALL
:TRIGGER:ATRIGGER:SPI:CLOCK:
POLARITY?
-> :TRIGGER:ATRIGGER:SPI:CLOCK:
POLARITY FALL

:TRIGger{[:ATRigger]][:BTRigger]:SPI:CLOCK:SOURce

Function Sets or queries the SPI bus signal trigger clock signal.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:SPI:CLOCK:SOURce {<NRF>|BIT<x>}
:TRIGger{[:ATRigger]][:BTRigger]:SPI:CLOCK:SOURce?
<NRF> = 1 to 4
BIT<x>'s <x> = 1 to 8

Example :TRIGGER:ATRIGGER:SPI:CLOCK:SOURCE 1
:TRIGGER:ATRIGGER:SPI:CLOCK:SOURCE?
-> :TRIGGER:ATRIGGER:SPI:CLOCK:
SOURCE 1

:TRIGger{[:ATRigger]][:BTRigger]:SPI:CS?

Function Queries all SPI bus signal trigger chip select signal settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:SPI:CS?

:TRIGger{[:ATRigger]][:BTRigger]:SPI:CS:ACTIVE

Function Sets or queries the active state of the SPI bus signal trigger chip select signal.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:SPI:CS:ACTIVE {HIGH|LOW}
:TRIGger{[:ATRigger]][:BTRigger]:SPI:CS:ACTIVE?

Example :TRIGGER:ATRIGGER:SPI:CS:ACTIVE HIGH
:TRIGGER:ATRIGGER:SPI:CS:ACTIVE?
-> :TRIGGER:ATRIGGER:SPI:CS:
ACTIVE HIGH

:TRIGger{[:ATRigger]][:BTRigger]:SPI:CS:SOURce

Function Sets or queries the SPI bus signal trigger chip select signal.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:SPI:CS:SOURce {<NRF>|BIT<x>}
:TRIGger{[:ATRigger]][:BTRigger]:SPI:CS:SOURce?
<NRF> = 1 to 4
BIT<x>'s <x> = 1 to 8

Example :TRIGGER:ATRIGGER:SPI:CS:SOURCE 1
:TRIGGER:ATRIGGER:SPI:CS:SOURCE?
-> :TRIGGER:ATRIGGER:SPI:CS:SOURCE 1

Description The chip select signal that you can specify varies depending on the :TRIGger{[:ATRigger]][:BTRigger]:SPI:CLOCK:SOURce setting. For details, see the Features Guide.

:TRIGger{[:ATRigger]][:BTRigger]:SPI:DATA<x>?

Function Queries all SPI bus signal trigger data signal settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:SPI:DATA<x>?
<x> = 1 to 2

:TRIGger{[:ATRigger]][:BTRigger]:SPI:DATA<x>:BCOUNT

Function Sets or queries the pattern comparison start position of the specified data for SPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:SPI:DATA<x>:BCOUNT {<NRF>}
:TRIGger{[:ATRigger]][:BTRigger]:SPI:DATA<x>:BCOUNT?
<x> = 1 to 2
<NRF> = 0 to 9999

Example :TRIGGER:ATRIGGER:SPI:DATA1:BCOUNT 0
:TRIGGER:ATRIGGER:SPI:DATA1:BCOUNT?
-> :TRIGGER:ATRIGGER:SPI:DATA1:
BCOUNT 0

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:CONDition

Function Sets or queries the judgment method (true/false) of the specified data for SPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:CONDition {FALSE|TRUE}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:CONDition?
<x> = 1 to 2

Example :TRIGGER:ATRIGGER:SPI:DATA1:
CONDITION FALSE
:TRIGGER:ATRIGGER:SPI:DATA1:
CONDITION?
-> :TRIGGER:ATRIGGER:SPI:DATA1:
CONDITION FALSE

:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:DBYTe

Function Sets or queries the data size (number of bytes) of the specified data for SPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:DBYTe {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:DBYTe?
<x> = 1 to 2
<Nrf> = 1 to 4

Example :TRIGGER:ATRIGGER:SPI:DATA1:DBYTE 1
:TRIGGER:ATRIGGER:SPI:DATA1:DBYTE?
-> :TRIGGER:ATRIGGER:SPI:DATA1:
DBYTE 1

:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:DPATtern?

Function Queries all SPI bus signal trigger data settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:DPATtern?
<x> = 1 to 2

:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:DPATtern:HEXa

Function Sets the specified data for SPI bus signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:DPATtern:HEXa {<String>}
<x> = 1 to 2

Example :TRIGGER:ATRIGGER:SPI:DATA1:
DPATTERN:HEXA "AB"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:DBYTe setting.

:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:DPATtern:PATtern

Function Sets or queries the specified data for SPI bus signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:DPATtern:PATtern {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:DPATtern:PATtern?
<x> = 1 to 2

Example :TRIGGER:ATRIGGER:SPI:DATA1:
DPATTERN:PATTERN "11001010"
:TRIGGER:ATRIGGER:SPI:DATA1:
DPATTERN:PATTERN?
-> :TRIGGER:ATRIGGER:SPI:DATA1:
DPATTERN:PATTERN "11001010"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:DBYTe setting.

:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:MODE

Function Sets or queries the enable/disable condition of data signal for SPI bus signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:MODE {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:MODE?
<x> = 1 to 2

Example :TRIGGER:ATRIGGER:SPI:DATA1:MODE ON
:TRIGGER:ATRIGGER:SPI:DATA1:MODE?
-> :TRIGGER:ATRIGGER:SPI:DATA1:
MODE 1

:TRIGger{[:ATRigger]|:BTRigger}:SPI:DATA<x>:SOURce

Function Sets or queries the SPI bus signal trigger data signal.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:SOURce {<Nrf>|BIT<y>}
:TRIGger{[:ATRigger]|:BTRigger}:SPI:
DATA<x>:SOURce?
<x> = 1 to 2
<Nrf> = 1 to 4
BIT<y>'s <y> = 1 to 8

Example :TRIGGER:ATRIGGER:SPI:DATA1:SOURCE 1
:TRIGGER:ATRIGGER:SPI:DATA1:SOURCE?
-> :TRIGGER:ATRIGGER:SPI:DATA1:
SOURCE 1

Description The data signal that you can specify varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:SPI:CLOCK:SOURce setting. For details, see the Features Guide.

:TRIGger{[:ATRigger]][:BTRigger]:SPI:MODE

Function Sets or queries the wiring system (three-wire system or four-wire system) of the SPI bus trigger.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:SPI:MODE {WIRE3|WIRE4}
:TRIGger{[:ATRigger]][:BTRigger]:SPI:MODE?

Example :TRIGGER:ATRIGGER:SPI:MODE WIRE3
:TRIGGER:ATRIGGER:SPI:MODE?
-> :TRIGGER:ATRIGGER:SPI:MODE WIRE3

:TRIGger{[:ATRigger]][:BTRigger]:TYPE

Function Sets or queries the trigger type.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:TYPE {INTERval|OR|PATTERN|PULSE|RFTIME|RUNT|SIMPLE|TIMEout|TV|WINDOW|WOR|CANBus|CANFDbus|CXPIbus|FLEXray|I2Cbus|LINbus|PSI5|SENT|SPATtern|SPIbus|UART}
:TRIGger{[:ATRigger]][:BTRigger]:TYPE?

Example :TRIGGER:ATRIGGER:TYPE SIMPLE
:TRIGGER:ATRIGGER:TYPE?
-> :TRIGGER:ATRIGGER:TYPE SIMPLE

Description The settings that are available for :TRIGger:BTRigger:TYPE are {OR|PATTERN|SIMPLE|WINDOW|WOR|CANBus|CANFDbus|CXPIbus|FLEXray|I2Cbus|LINbus|SENT|SPATtern|SPIbus|UART}.

:TRIGger{[:ATRigger]][:BTRigger]:UART?

Function Queries all UART signal trigger settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:UART?

:TRIGger{[:ATRigger]][:BTRigger]:UART:BITOrder

Function Sets or queries the UART signal trigger bit order.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:UART:BITOrder {LSBFirst|MSBFirst}
:TRIGger{[:ATRigger]][:BTRigger]:UART:BITOrder?

Example :TRIGGER:ATRIGGER:UART:BITORDER LSBFIRST
:TRIGGER:ATRIGGER:UART:BITORDER?
-> :TRIGGER:ATRIGGER:UART:BITORDER LSBFIRST

:TRIGger{[:ATRigger]][:BTRigger]:UART:BRATe

Function Sets or queries the UART signal trigger bit rate (data transfer rate).

Syntax :TRIGger{[:ATRigger]][:BTRigger]:UART:BRATe {<Nrf>|USER,<Nrf>}
:TRIGger{[:ATRigger]][:BTRigger]:UART:BRATe?<Nrf> = 1200, 2400, 4800, 9600, 19200
USER <Nrf> = See the Features Guide for this information.

Example :TRIGGER:ATRIGGER:UART:BRATE 2400
:TRIGGER:ATRIGGER:UART:BRATE?
-> :TRIGGER:ATRIGGER:UART:BRATE 2400

:TRIGger{[:ATRigger]][:BTRigger]:UART:DATA?

Function Queries all UART signal trigger data settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:UART:DATA?

:TRIGger{[:ATRigger]][:BTRigger]:UART:DATA:ASCII

Function Sets the UART signal trigger data in ASCII format.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:UART:DATA:ASCII {<String>}

Example :TRIGGER:ATRIGGER:UART:DATA:ASCII "TEST"

:TRIGger{[:ATRigger]][:BTRigger]:UART:DATA:CONDITION

Function Sets or queries the data judgment method (true/false) for UART signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:UART:DATA:CONDITION {DONTcare|TRUE}
:TRIGger{[:ATRigger]][:BTRigger]:UART:DATA:CONDITION?

Example :TRIGGER:ATRIGGER:UART:DATA:CONDITION DONTCARE
:TRIGGER:ATRIGGER:UART:DATA:CONDITION?
-> :TRIGGER:ATRIGGER:UART:DATA:CONDITION DONTCARE

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:CSensitive

Function Sets or queries whether to distinguish uppercase and lowercase letters in ASCII data for UART signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:CSensitive {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:CSensitive?

Example :TRIGGER:ATRIGGER:UART:DATA:
CSENSITIVE ON
:TRIGGER:ATRIGGER:UART:DATA:
CSENSITIVE?
-> :TRIGGER:ATRIGGER:UART:DATA:
CSENSITIVE 1

:TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:DBYTE

Function Sets or queries the number of data bytes for UART signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:DBYTE {<NRF>}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:DBYTE?
<NRF> = 1 to 4(byte)

Example :TRIGGER:ATRIGGER:UART:DATA:DBYTE 1
:TRIGGER:ATRIGGER:UART:DATA:DBYTE?
-> :TRIGGER:ATRIGGER:UART:DATA:
DBYTE 1

:TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:DPATtern?

Function Queries all UART signal trigger data settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:DPATtern?

:TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:DPATtern:HEXa

Function Sets the data for UART signal triggering in hexadecimal notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:DPATtern:HEXa {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:DPATtern:HEXa?

Example :TRIGGER:ATRIGGER:UART:DATA:
DPATTERN:HEXa "12"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:DBYTE? setting.

:TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:DPATtern:PATtern

Function Sets or queries the data for UART signal triggering in binary notation.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:DPATtern:PATtern {<String>}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:DPATtern:PATtern?

Example :TRIGGER:ATRIGGER:UART:DATA:
DPATTERN:PATTERN "00110101"
:TRIGGER:ATRIGGER:UART:DATA:
DPATTERN:PATTERN?
-> :TRIGGER:ATRIGGER:UART:DATA:
DPATTERN:PATTERN "00110101"

Description The number of bytes that can be set varies depending on the :TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:DBYTE? setting.

:TRIGger{[:ATRigger]|:BTRigger}:UART:DATA:PFORMAT

Function Sets or queries the data input format, which is one of the ID and data conditions, for UART signal triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:PFORMAT {ASCII|PATtern}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:DATA:PFORMAT?

Example :TRIGGER:ATRIGGER:UART:DATA:
PFORMAT PATTERN
:TRIGGER:ATRIGGER:UART:DATA:PFORMAT?
-> :TRIGGER:ATRIGGER:UART:DATA:
PFORMAT PATTERN

:TRIGger{[:ATRigger]|:BTRigger}:UART:ERROR?

Function Queries all UART signal trigger error settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
UART:ERROR?

:TRIGger{[:ATRigger]|:BTRigger}:UART:ERROR:FRAMing

Function Sets or queries the UART signal trigger framing error setting.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
UART:ERROR:FRAMing {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:
UART:ERROR:FRAMing?

Example :TRIGGER:ATRIGGER:UART:ERROR:
FRAMING ON
:TRIGGER:ATRIGGER:UART:ERROR:
FRAMING?
-> :TRIGGER:ATRIGGER:UART:ERROR:
FRAMING 1

**:TRIGger{[:ATRigger]][:BTRigger]:UART:ER
Ror:PARity**

Function Sets or queries the UART signal trigger parity error setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
UART:ERror:PARity {<Boolean>}
:TRIGger{[:ATRigger]][:BTRigger]:
UART:ERror:PARity?

Example :TRIGGER:ATRIGGER:UART:ERROR:
PARITY ON
:TRIGGER:ATRIGGER:UART:ERROR:PARITY?
-> :TRIGGER:ATRIGGER:UART:ERROR:
PARITY 1

**:TRIGger{[:ATRigger]][:BTRigger]:UART:ER
Ror:PMODE**

Function Sets or queries the UART signal trigger parity mode setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
UART:ERror:PMODE {EVEN|ODD}
:TRIGger{[:ATRigger]][:BTRigger]:
UART:ERror:PMODE?

Example :TRIGGER:ATRIGGER:UART:ERROR:
PMODE EVEN
:TRIGGER:ATRIGGER:UART:ERROR:PMODE?
-> :TRIGGER:ATRIGGER:UART:ERROR:
PMODE EVEN

**:TRIGger{[:ATRigger]][:BTRigger]:UART:FO
Rmat**

Function Sets or queries the UART signal trigger format setting.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
UART:FORmat {BIT7Parity|BIT8Parity|
BIT8Noparity}
:TRIGger{[:ATRigger]][:BTRigger]:
UART:FORmat?

Example :TRIGGER:ATRIGGER:UART:
FORMAT BIT7PARITY
:TRIGGER:ATRIGGER:UART:FORMAT?
-> :TRIGGER:ATRIGGER:UART:
FORMAT BIT7PARITY

**:TRIGger{[:ATRigger]][:BTRigger]:UART:MO
DE**

Function Sets or queries the trigger type for UART signal triggering.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
UART:MODE {DATA|EDaTa|ERror}
:TRIGger{[:ATRigger]][:BTRigger]:
UART:MODE?

Example :TRIGGER:ATRIGGER:UART:MODE DATA
:TRIGGER:ATRIGGER:UART:MODE?
-> :TRIGGER:ATRIGGER:UART:MODE DATA

**:TRIGger{[:ATRigger]][:BTRigger]:UART:PO
Larity**

Function Sets or queries the UART signal analysis polarity.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
UART:POLarity {NEGative|POSitive}
:TRIGger{[:ATRigger]][:BTRigger]:
UART:POLarity?

Example :TRIGGER:ATRIGGER:UART:
POLARITY NEGATIVE
:TRIGGER:ATRIGGER:UART:POLARITY?
-> :TRIGGER:ATRIGGER:UART:
POLARITY NEGATIVE

**:TRIGger{[:ATRigger]][:BTRigger]:UART:SO
URce**

Function Sets or queries the UART signal analysis signal.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
UART:SOURce {<NRF>|BIT<x>}
:TRIGger{[:ATRigger]][:BTRigger]:
UART:SOURce?

<NRF> = 1 to 4
BIT<x>'s <x> = 1 to 8
Example :TRIGGER:ATRIGGER:UART:SOURCE 1
:TRIGGER:ATRIGGER:UART:SOURCE?
-> :TRIGGER:ATRIGGER:UART:SOURCE 1

**:TRIGger{[:ATRigger]][:BTRigger]:UART:SP
Oint**

Function Sets or queries the UART signal analysis sample point.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
UART:SPOint {<NRF>}
:TRIGger{[:ATRigger]][:BTRigger]:
UART:SPOint?

<NRF> = 18.8 to 90.6
Example :TRIGGER:ATRIGGER:UART:SPOINT 18.8
:TRIGGER:ATRIGGER:UART:SPOINT?
-> :TRIGGER:ATRIGGER:UART:
SPOINT 18.8

:TRIGger{[:ATRigger]][:BTRigger]:WINDow?

Function Queries all window trigger settings.

Syntax :TRIGger{[:ATRigger]][:BTRigger]:
WINDow?

5.33 TRIGger Group

:TRIGger{[:ATRigger]|:BTRigger}:WINDOW:POLarity

Function Sets or queries the polarity of the window trigger source.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
WINDOW:POLarity {IN|OUT}
:TRIGger{[:ATRigger]|:BTRigger}:
WINDOW:POLarity?

Example :TRIGGER:ATRIGGER:WINDOW:POLARITY IN
:TRIGGER:ATRIGGER:WINDOW:POLARITY?
-> :TRIGGER:ATRIGGER:WINDOW:
POLARITY IN

:TRIGger{[:ATRigger]|:BTRigger}:WINDOW:SOURce

Function Sets or queries the window trigger source.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
WINDOW:SOURce {<Nrf>}
:TRIGger{[:ATRigger]|:BTRigger}:
WINDOW:SOURce?
<Nrf> = 1 to 4

Example :TRIGGER:ATRIGGER:WINDOW:SOURCE 1
:TRIGGER:ATRIGGER:WINDOW:SOURCE?
-> :TRIGGER:ATRIGGER:WINDOW:SOURCE 1

:TRIGger{[:ATRigger]|:BTRigger}:WINDOW:TQQualify

Function Sets or queries the time condition for window triggering.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:
WINDOW:TQQualify {INSide|LTHan|MTHan|
NONE|OUTSide|TIMEout}
:TRIGger{[:ATRigger]|:BTRigger}:
WINDOW:TQQualify?

Example :TRIGGER:ATRIGGER:WINDOW:
TQUALIFY INSIDE
:TRIGGER:ATRIGGER:WINDOW:TQUALIFY?
-> :TRIGGER:ATRIGGER:WINDOW:
TQUALIFY INSIDE

:TRIGger{[:ATRigger]|:BTRigger}:WOR?

Function Queries all window OR trigger settings.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:WOR?

:TRIGger{[:ATRigger]|:BTRigger}:WOR:ALL

Function Sets or queries whether to make all channels window OR trigger sources at once.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:WOR:
ALL {<Boolean>}
:TRIGger{[:ATRigger]|:BTRigger}:WOR:
ALL?

Example :TRIGGER:ATRIGGER:WOR:ALL ON
:TRIGGER:ATRIGGER:WOR:ALL?
-> :TRIGGER:ATRIGGER:WOR:ALL 1

:TRIGger{[:ATRigger]|:BTRigger}:WOR:CHANnel<x>

Function Sets or queries the slope of each channel of the window OR trigger.

Syntax :TRIGger{[:ATRigger]|:BTRigger}:WOR:
CHANnel<x> {IN|OFF|OUT}
:TRIGger{[:ATRigger]|:BTRigger}:WOR:
CHANnel<x>?
<x> = 1 to 4

Example :TRIGGER:ATRIGGER:WOR:CHANNEL1 IN
:TRIGGER:ATRIGGER:WOR:CHANNEL1?
-> :TRIGGER:ATRIGGER:WOR:CHANNEL1 IN

:TRIGger[:ATRigger]:INTERval?

Function Queries all edge interval trigger settings.

Syntax :TRIGger[:ATRigger]:INTERval?

:TRIGger[:ATRigger]:INTERval:SLOPe

Function Sets or queries the edge interval trigger slope.

Syntax :TRIGger[:ATRigger]:INTERval:
SLOPe {FALL|RISE}
:TRIGger[:ATRigger]:INTERval:SLOPe?

Example :TRIGGER:ATRIGGER:INTERVAL:
SLOPE FALL
:TRIGGER:ATRIGGER:INTERVAL:SLOPE?
-> :TRIGGER:ATRIGGER:INTERVAL:
SLOPE FALL

:TRIGger[:ATRigger]:INTERval:SOURce

Function Sets or queries the edge interval trigger source.

Syntax :TRIGger[:ATRigger]:INTERval:
SOURce <Nrf>|BIT<x>}
:TRIGger[:ATRigger]:INTERval:SOURce?
<Nrf> = 1 to 4
BIT<x>'s <x> = 1 to 8

Example :TRIGGER:ATRIGGER:INTERVAL:SOURCE 1
:TRIGGER:ATRIGGER:INTERVAL:SOURCE?
-> :TRIGGER:ATRIGGER:INTERVAL:
SOURCE 1

:TRIGger[:ATRigger]:INTERval:TQQualify

Function Sets or queries the time condition for edge interval triggering.

Syntax :TRIGger[:ATRigger]:INTERval:
TQQualify {INSide|LTHan|MTHan|
OUTSide|TIMEout}
:TRIGger[:ATRigger]:INTERval:
TQQualify?

Example :TRIGGER:ATRIGGER:INTERVAL:
TQUALIFY INSIDE
:TRIGGER:ATRIGGER:INTERVAL:TQUALIFY?
-> :TRIGGER:ATRIGGER:INTERVAL:
TQUALIFY INSIDE

:TRIGger[:ATRigger]:PULSe?

Function Queries all pulse width trigger settings.

Syntax :TRIGger[:ATRigger]:PULSe?

:TRIGger[:ATRigger]:PULSe:POLarity

Function Sets or queries the pulse width trigger source polarity.

Syntax :TRIGger[:ATRigger]:PULSe:
POLarity {NEGative|POSitive}
:TRIGger[:ATRigger]:PULSe:POLarity?

Example :TRIGGER:ATRIGGER:PULSE:
POLARITY NEGATIVE
:TRIGGER:ATRIGGER:PULSE:POLARITY?
-> :TRIGGER:ATRIGGER:PULSE:
POLARITY NEGATIVE

:TRIGger[:ATRigger]:PULSe:SOURce

Function Sets or queries the pulse width trigger source.

Syntax :TRIGger[:ATRigger]:PULSe:
SOURce {<Nrf>|BIT<x>}
:TRIGger[:ATRigger]:PULSe:SOURce?
<Nrf> = 1 to 4
BIT<x>'s <x> = 1 to 8

Example :TRIGGER:ATRIGGER:PULSE:SOURCE 1
:TRIGGER:ATRIGGER:PULSE:SOURCE?
-> :TRIGGER:ATRIGGER:PULSE:SOURCE 1

:TRIGger[:ATRigger]:PULSe:TQQualify

Function Sets or queries the time condition for pulse width triggering.

Syntax :TRIGger[:ATRigger]:PULSe:
TQQualify {INSide|LTHan|MTHan|
OUTSide|TIMEout}
:TRIGger[:ATRigger]:PULSe:TQQualify?

Example :TRIGGER:ATRIGGER:PULSE:
TQUALIFY INSIDE
:TRIGGER:ATRIGGER:PULSE:TQUALIFY?
-> :TRIGGER:ATRIGGER:PULSE:
TQUALIFY INSIDE

:TRIGger[:ATRigger]:RFTTime?

Function Queries all rise/fall time trigger settings.

Syntax :TRIGger[:ATRigger]:RFTTime?

:TRIGger[:ATRigger]:RFTTime:SLOPe

Function Sets or queries the rise/fall time trigger slope.

Syntax :TRIGger[:ATRigger]:RFTTime:
SLOPe {BOTH|FALL|RISE}
:TRIGger[:ATRigger]:RFTTime:SLOPe?

Example :TRIGGER:ATRIGGER:RFTIME:SLOPE FALL
:TRIGGER:ATRIGGER:RFTIME:SLOPE?
-> :TRIGGER:ATRIGGER:RFTIME:
SLOPE FALL

:TRIGger[:ATRigger]:RFTTime:SOURce

Function Sets or queries the rise/fall time trigger source.

Syntax :TRIGger[:ATRigger]:RFTTime:
SOURce {<Nrf>}
:TRIGger[:ATRigger]:RFTTime:SOURce?
<Nrf> = 1 to 4

Example :TRIGGER:ATRIGGER:RFTIME:SOURCE 1
:TRIGGER:ATRIGGER:RFTIME:SOURCE?
-> :TRIGGER:ATRIGGER:RFTIME:SOURCE 1

:TRIGger[:ATRigger]:RFTTime:TQQualify

Function Sets or queries the time condition for rise/fall time triggering.

Syntax :TRIGger[:ATRigger]:RFTTime:
TQQualify {INSide|LTHan|MTHan|
OUTSide}
:TRIGger[:ATRigger]:RFTTime:TQQualify?

Example :TRIGGER:ATRIGGER:RFTIME:
TQUALIFY INSIDE
:TRIGGER:ATRIGGER:RFTIME:TQUALIFY?
-> :TRIGGER:ATRIGGER:RFTIME:
TQUALIFY INSIDE

5.33 TRIGger Group

:TRIGger[:ATRigger]:RUNT?

Function Queries all runt interval trigger settings.
Syntax :TRIGger[:ATRigger]:RUNT?

:TRIGger[:ATRigger]:RUNT:POLarity

Function Sets or queries the polarity of the runt trigger source.

Syntax :TRIGger[:ATRigger]:RUNT:
POLarity {EITHer|NEGative|POSitive}
:TRIGger[:ATRigger]:RUNT:POLarity?

Example :TRIGGER:ATRIGGER:RUNT:
POLARITY NEGATIVE
:TRIGGER:ATRIGGER:RUNT:POLARITY?
-> :TRIGGER:ATRIGGER:RUNT:
POLARITY NEGATIVE

:TRIGger[:ATRigger]:RUNT:SOURce

Function Sets or queries the runt trigger source.

Syntax :TRIGger[:ATRigger]:RUNT:
SOURce {<NRf>}
:TRIGger[:ATRigger]:RUNT:SOURce?
<NRf> = 1 to 4

Example :TRIGGER:ATRIGGER:RUNT:SOURCE 1
:TRIGGER:ATRIGGER:RUNT:SOURCE?
-> :TRIGGER:ATRIGGER:RUNT:SOURCE 1

:TRIGger[:ATRigger]:RUNT:TQQualify

Function Sets or queries the time condition for runt triggering.

Syntax :TRIGger[:ATRigger]:RUNT:
TQQualify {INSide|LTHan|MTHan|NONE|
OUTSide}
:TRIGger[:ATRigger]:RUNT:TQQualify?

Example :TRIGGER:ATRIGGER:RUNT:
TQUALIFY INSIDE
:TRIGGER:ATRIGGER:RUNT:TQUALIFY?
-> :TRIGGER:ATRIGGER:RUNT:
TQUALIFY INSIDE

:TRIGger[:ATRigger]:TIMEout?

Function Queries all timeout trigger settings.
Syntax :TRIGger[:ATRigger]:TIMEout?

:TRIGger[:ATRigger]:TIMEout:POLarity

Function Sets or queries the polarity of the timeout period trigger source.

Syntax :TRIGger[:ATRigger]:TIMEout:
POLarity {EITHer|NEGative|POSitive}
:TRIGger[:ATRigger]:TIMEout:
POLarity?

Example :TRIGGER:ATRIGGER:TIMEOUT:
POLARITY NEGATIVE
:TRIGGER:ATRIGGER:TIMEOUT:POLARITY?
-> :TRIGGER:ATRIGGER:TIMEOUT:
POLARITY NEGATIVE

:TRIGger[:ATRigger]:TIMEout:SOURce

Function Sets or queries the timeout period trigger source.

Syntax :TRIGger[:ATRigger]:TIMEout:
SOURce {<NRf>|BIT<x>}
:TRIGger[:ATRigger]:TIMEout:SOURce?
<NRf> = 1 to 4
BIT<x>'s <x> = 1 to 8

Example :TRIGGER:ATRIGGER:TIMEOUT:SOURCE 1
:TRIGGER:ATRIGGER:TIMEOUT:SOURCE?
-> :TRIGGER:ATRIGGER:TIMEOUT:
SOURCE 1

:TRIGger[:ATRigger]:TV?

Function Queries all TV trigger settings.
Syntax :TRIGger[:ATRigger]:TV?

:TRIGger[:ATRigger]:TV:FIELD

Function Sets or queries the field that the TV trigger is to be activated on.

Syntax :TRIGger[:ATRigger]:TV:
FIELD {DONTcare|<NRf>}
:TRIGger[:ATRigger]:TV:FIELD?
<NRf> = 1, 2

Example :TRIGGER:ATRIGGER:TV:FIELD 1
:TRIGGER:ATRIGGER:TV:FIELD?
-> :TRIGGER:ATRIGGER:TV:FIELD 1

:TRIGger[:ATRigger]:TV:FRAME

Function Sets or queries the TV trigger frame skip function.

Syntax :TRIGger[:ATRigger]:TV:FRAME {<NRf>}
:TRIGger[:ATRigger]:TV:FRAME?
<NRf> = 1, 2, 4, 8

Example :TRIGGER:ATRIGGER:TV:FRAME 2
:TRIGGER:ATRIGGER:TV:FRAME?
-> :TRIGGER:ATRIGGER:TV:FRAME 2

:TRIGger[:ATRigger]:TV:{HDTV|NTSC|PAL|SDTV|USERdefine}?

Function Queries all TV trigger mode settings.

Syntax :TRIGger[:ATRigger]:TV:{HDTV|NTSC|PAL|SDTV|USERdefine}?

:TRIGger[:ATRigger]:TV:{HDTV|NTSC|PAL|SDTV|USERdefine}:LINE

Function Sets or queries the line that the TV trigger is to be activated on.

Syntax :TRIGger[:ATRigger]:TV:{HDTV|NTSC|PAL|SDTV|USERdefine}:LINE {<Nrf>}
:TRIGger[:ATRigger]:TV:{HDTV|NTSC|PAL|SDTV|USERdefine}:LINE?
<Nrf> = 2 to 2251

Example The following example is for HDTV.

```
:TRIGGER:ATRIGGER:TV:HDTV:LINE 2
:TRIGGER:ATRIGGER:TV:HDTV:LINE?
-> :TRIGGER:ATRIGGER:TV:HDTV:LINE 2
```

:TRIGger[:ATRigger]:TV:{HDTV|NTSC|PAL|SDTV|USERdefine}:POLarity

Function Sets or queries the TV trigger input polarity.

Syntax :TRIGger[:ATRigger]:TV:HDTV:POLarity {POSitive|NEGative}
:TRIGger[:ATRigger]:TV:HDTV:POLarity?

Example The following example is for HDTV.

```
:TRIGGER:ATRIGGER:TV:HDTV:POLARITY POSITIVE
:TRIGGER:ATRIGGER:TV:HDTV:POLARITY?
-> :TRIGGER:ATRIGGER:TV:HDTV:POLARITY POSITIVE
```

:TRIGger[:ATRigger]:TV:LEVel

Function Sets or queries the TV trigger level.

Syntax :TRIGger[:ATRigger]:TV:LEVel {<Nrf>}
:TRIGger[:ATRigger]:TV:LEVel?
<Nrf> = 0.1 div to 2.0 div, 0.1 div steps

Example :TRIGGER:ATRIGGER:TV:LEVEL 0.1

```
:TRIGGER:ATRIGGER:TV:LEVEL?
-> :TRIGGER:ATRIGGER:TV:LEVEL 0.1
```

:TRIGger[:ATRigger]:TV:LFORmat

Function Sets or queries the line designation method for TV triggering.

Syntax :TRIGger[:ATRigger]:TV:LFORmat {ALL|ONE}
:TRIGger[:ATRigger]:TV:LFORmat?

Example :TRIGGER:ATRIGGER:TV:LFORMAT ALL

```
:TRIGGER:ATRIGGER:TV:LFORMAT?
```

```
-> :TRIGGER:ATRIGGER:TV:LFORMAT ALL
```

:TRIGger[:ATRigger]:TV:SOURce

Function Sets or queries the TV trigger source.

Syntax :TRIGger[:ATRigger]:TV:SOURce {<Nrf>}
:TRIGger[:ATRigger]:TV:SOURce?
<Nrf> = 1 to 4

Example :TRIGGER:ATRIGGER:TV:SOURCE 1

```
:TRIGGER:ATRIGGER:TV:SOURCE?
```

```
-> :TRIGGER:ATRIGGER:TV:SOURCE 1
```

:TRIGger[:ATRigger]:TV:TYPE

Function Sets or queries the TV trigger input type.

Syntax :TRIGger[:ATRigger]:TV:TYPE {NTSC|HDTV[, {I1080 _ 50|I1080 _ 60|P1080 _ 24|P1080 _ 25|P1080 _ 60|P720 _ 60|SF1080 _ 24}]|PAL|SDTV|USERdefine}
:TRIGger[:ATRigger]:TV:TYPE?

Example :TRIGGER:ATRIGGER:TV:TYPE NTSC

```
:TRIGGER:ATRIGGER:TV:TYPE?
```

```
-> :TRIGGER:ATRIGGER:TV:TYPE NTSC
```

:TRIGger[:ATRigger]:TV:USERdefine?

Function Queries all user-defined TV trigger settings.

Syntax :TRIGger[:ATRigger]:TV:USERdefine?

:TRIGger[:ATRigger]:TV:USERdefine:DEFinition

Function Sets or queries the user-defined TV trigger definition.

Syntax :TRIGger[:ATRigger]:TV:USERdefine:DEFinition {HD|SD}
:TRIGger[:ATRigger]:TV:USERdefine:DEFinition?

Example :TRIGGER:ATRIGGER:TV:USERDEFINE:

```
DEFINITION HD
```

```
:TRIGGER:ATRIGGER:TV:USERDEFINE:
```

```
DEFINITION?
```

```
-> :TRIGGER:ATRIGGER:TV:USERDEFINE:
```

```
DEFINITION HD
```

5.33 TRIGger Group

:TRIGger[:ATRigger]:TV:USERdefine:HFRejection (HighFrequencyREJECTION)

Function Sets or queries the on/off state of the low-pass filter (HF rejection) for user-defined TV triggering.

Syntax :TRIGger[:ATRigger]:TV:USERdefine:
HFRejection {<Frequency>|OFF}
:TRIGger[:ATRigger]:TV:USERdefine:
HFRejection?

Example :TRIGGER:ATRIGGER:TV:USERDEFINE:
HFREJECTION 300KHZ
:TRIGGER:ATRIGGER:TV:USERDEFINE:
HFREJECTION?
-> :TRIGGER:ATRIGGER:TV:USERDEFINE:
HFREJECTION 300000

:TRIGger[:ATRigger]:TV:USERdefine:HSYNc (Hsync Freq)

Function Sets or queries the horizontal sync signal frequency for user-defined TV triggering.

Syntax :TRIGger[:ATRigger]:TV:USERdefine:
HSYNc {<Frequency>}
:TRIGger[:ATRigger]:TV:USERdefine:
HSYNc?

Example :TRIGGER:ATRIGGER:TV:USERDEFINE:
HSYNC 10KHZ
:TRIGGER:ATRIGGER:TV:USERDEFINE:
HSYNC?
-> :TRIGGER:ATRIGGER:TV:USERDEFINE:
HSYNC 10.000E+03

:TRIGger[:ATRigger]:TV:USERdefine:SGUard

Function Sets or queries the sync guard for user-defined TV triggering.

Syntax :TRIGger[:ATRigger]:TV:USERdefine:
SGUard {<NRf>}
:TRIGger[:ATRigger]:TV:USERdefine:
SGUard?

Example :TRIGGER:ATRIGGER:TV:USERDEFINE:
SGUARD 60
:TRIGGER:ATRIGGER:TV:USERDEFINE:
SGUARD?
-> :TRIGGER:ATRIGGER:TV:USERDEFINE:
SGUARD 60

:TRIGger:COMBination

Function Sets or queries the trigger combination.

Syntax :TRIGger:COMBination {ABN|ADB|OFF}
:TRIGger:COMBination?

Example :TRIGGER:COMBINATION ABN
:TRIGGER:COMBINATION?
-> :TRIGGER:COMBINATION ABN

:TRIGger:DElay?

Function Queries all delay settings.

Syntax :TRIGger:DElay?

:TRIGger:DElay:TIME

Function Sets or queries the delay (time from the trigger point to the trigger position).

Syntax :TRIGger:DElay:TIME {<Time>}
:TRIGger:DElay:TIME?
<Time> = See the Features Guide for this information.

Example :TRIGGER:DELAY:TIME 2US
:TRIGGER:DELAY:TIME?
-> :TRIGGER:DELAY:TIME 2.00E-06

Description The value is fixed to 0 when the time base is set to external clock. On this instrument, set the trigger delay using the time from the trigger point to the trigger position. Use :TRIGger:POSition to set or query the trigger position.

:TRIGger:DETEct?

Function Queries whether a trigger has occurred.

Syntax :TRIGger:DETEct? [{<NRf>}]
 <NRf> = 0 to 360000, timeout value, 10 ms steps

Example :TRIGGER:DETEct? 0
 -> :TRIGGER:DETEct 1

Description

- The instrument returns 1 when it is triggered, 0 when it is not. If the instrument was triggered before sending this command, it returns 1.
- If a timeout period is set, the instrument returns 1 when it is triggered. If the instrument is not triggered within the timeout period, it returns 0. If the timeout period is set to 0, the instrument returns 1 if it is triggered when the command is sent or 0 otherwise.
- The timeout period can be omitted. If omitted, the behavior is the same as when <NRf>=0.
- When the following operation is performed, the instrument resets to the state where no trigger is detected (the state where the instrument returns 0 when this command is sent).
 - When this command is sent
 - When waveform acquisition starts
 - When the waveform acquisition conditions are changed while waveform acquisition is in progress

:TRIGger:FORCE

Function Forces the instrument to trigger.

Syntax :TRIGger:FORCE

Example :TRIGGER:FORCE

:TRIGger:HOLDoff?

Function Queries all hold-off settings.

Syntax :TRIGger:HOLDoff?

:TRIGger:HOLDoff:TIME

Function Sets or queries the hold-off time.

Syntax :TRIGger:HOLDoff:TIME {<Time>}
 :TRIGger:HOLDoff:TIME?
 <Time> = 20 ns to 10 s

Example :TRIGGER:HOLDOFF:TIME 20NS
 :TRIGGER:HOLDOFF:TIME?
 -> :TRIGGER:HOLDOFF:TIME 20.0E-09

:TRIGger:MODE

Function Sets or queries the trigger mode.

Syntax :TRIGger:MODE {AUTO|ALEVel|NORMal|NSINgle}
 :TRIGger:MODE?

Example :TRIGGER:MODE AUTO
 :TRIGGER:MODE?
 -> :TRIGGER:MODE AUTO

Description Use :TRIGger:SCount to set or query the count for NSINgle.

:TRIGger:POSition

Function Sets or queries the trigger position.

Syntax :TRIGger:POSition {<NRf>}
 :TRIGger:POSition?
 <NRf> = 0% to 100%, 0.1% steps

Example :TRIGGER:POSITION 50
 :TRIGGER:POSITION?
 -> :TRIGGER:POSITION 50.000

:TRIGger:SCount

Function Sets or queries the number of times the trigger condition is to be met when the trigger mode is set to N Single.

Syntax :TRIGger:SCount {<NRf>}
 :TRIGger:SCount?
 <NRf> = 1 to 100000

Example :TRIGGER:SCOUNT 100
 :TRIGGER:SCOUNT?
 -> :TRIGGER:SCOUNT 100

:TRIGger:SOURce?

Function Queries all enhanced trigger source settings.

Syntax :TRIGger:SOURce?

:TRIGger:SOURce:CHANnel<x>?

Function Queries all settings related to the specified trigger source for enhanced triggering.

Syntax :TRIGger:SOURce:CHANnel<x>?
 <x> = 1 to 4

5.33 TRIGger Group

:TRIGger:SOURce:CHANnel<x>:HFRejection (HighFrequencyREJECTION)

Function Sets or queries the low-pass filter of the specified trigger source for enhanced triggering.

Syntax :TRIGger:SOURce:CHANnel<x>:
HFRejection {<Frequency>|OFF}
:TRIGger:SOURce:CHANnel<x>:
HFRejection?

<x> = 1 to 4
Frequency: 15 kHz to 20 MHz

Example :TRIGGER:SOURCE:CHANNEL1:
HFREJECTION 15KHZ
:TRIGGER:SOURCE:CHANNEL1:
HFREJECTION?
-> :TRIGGER:SOURCE:CHANNEL1:
HFREJECTION 15000

:TRIGger:SOURce:CHANnel<x>:LEVel

Function Sets or queries the trigger level of the specified trigger source for enhanced triggering.

Syntax :TRIGger:SOURce:CHANnel<x>:
LEVel {<Voltage>}
:TRIGger:SOURce:CHANnel<x>:LEVel?
<x> = 1 to 4
<Voltage> = See the Features Guide for this information.

Example :TRIGGER:SOURCE:CHANNEL1:LEVEL 0V
:TRIGGER:SOURCE:CHANNEL1:LEVEL?
-> :TRIGGER:SOURCE:CHANNEL1:
LEVEL 0.000E+00

Description When a current probe is in use, this command does not set or query the <Current> value.

:TRIGger:SOURce:CHANnel<x>:NREJection

Function Sets or queries the noise rejection of the specified trigger source for enhanced triggering.

Syntax :TRIGger:SOURce:CHANnel<x>:
NREJection {HIGH|LOW|MIDDLE}
:TRIGger:SOURce:CHANnel<x>:
NREJection?
<x> = 1 to 4

Example :TRIGGER:SOURCE:CHANNEL1:
NREJECTION HIGH
:TRIGGER:SOURCE:CHANNEL1:NREJECTION?
-> :TRIGGER:SOURCE:CHANNEL1:
NREJECTION HIGH

:TRIGger:SOURce:CHANnel<x>:ULLeVel

Function Sets or queries the upper and lower limits of the enhanced trigger source level.

Syntax :TRIGger:SOURce:CHANnel<x>:
ULLeVel {<Voltage>,<Voltage>}
:TRIGger:SOURce:CHANnel<x>:ULLeVel?
<x> = 1 to 4
<Voltage> = See the Features Guide for this information.

Example :TRIGGER:SOURCE:CHANNEL1:
ULLEVEL 0.5V,-0.5V
:TRIGGER:SOURCE:CHANNEL1:ULLEVEL?
-> :TRIGGER:SOURCE:CHANNEL1:
ULLEVEL 0.500E+00,-0.500E+00

:TRIGger:TIME<x>

Function Sets or queries the time for enhanced triggering.

Syntax :TRIGger:TIME<x> {<Time>}
:TRIGger:TIME<x>?
<x> = 1 to 2
<Time> = 4 ns to 10 s (2 ns steps)

Example :TRIGGER:TIME1 1S
:TRIGGER:TIME1?
-> :TRIGGER:TIME1 1S

5.34 WAVEform Group

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

:WAVEform?

Function Queries all waveform data output settings.
Syntax :WAVEform?

:WAVEform:ALL?

Function Queries all settings related to the :WAVEform:ALL:SEND? query.
Syntax :WAVEform:ALL?

:WAVEform:ALL:SEND?

Function Queries the waveform data specified by :WAVEform:ALL:TRACe.

Syntax :WAVEform:ALL:SEND? {<NRf>}
<NRf>= 0 to the acquisition count

Example :WAVEFORM:ALL:SEND? 0
-> #800007558
03 00 85 00 00 00 00 00 00 00 00 00 00 00 00 E2 04
00 00 00 00 00 00 E2 04 00 00 62 00 AC 11 44 00
6C 00 26 00 08 00 3C 18 3E 18 4E 18 0A 18 18 ...

Description

- You can query waveform data during measurement.
- You cannot query waveform data when the trigger mode is set to Single or NSingle.
- You cannot query waveform data when the display is in roll mode.
- You cannot query waveform data when continuous statistical processing of automatically measured waveform parameters or user-defined computation is enabled.
- The data format is shown below.
#8 (eight digits for the number of transmission data bytes in decimal format) <- Block data header
(two bytes for the number of traces)(eight bytes for the acquisition count) <- Common information
(four bytes for the trace number)(eight bytes for the reserved area)(four bytes for the number of data points) <- Trace information 1
(data sequence) . . .
(four bytes for the trace number)(eight bytes for the reserved area)(four bytes for the number of data points) <- Trace information 2
(data sequence) . . .
. . . (repeated for all the traces)
- The common information and trace information is written in little-endian byte order.
- The data sequence format is WORD. The byte order is little endian.
- You can query up to 12.5 kPoints in a record.
- You can acquire the latest waveform data by setting <NRf> (the acquisition count) to 0.
- To acquire waveforms so that the same waveform data is not acquired, specify the acquisition count of the common information acquired with this command the last time using <NRf>. If the specified acquisition count and the instrument's acquisition count are the same at the point in time that the query is received, #800000000 is returned.

5.34 WAVEform Group

:WAVEform:ALL:TRACe

Function Sets or queries the target waveform for the :WAVEform:ALL:SEND? command.

Syntax :WAVEform:ALL:TRACe {<NRf>|ALL|LOGic|MATH<x>}:WAVEform:ALL:TRACe?<NRf> = 1 to 4<x> = 1 to 4

Example :WAVEFORM:ALL:TRACE 1
:WAVEFORM:ALL:TRACE?
-> :WAVEFORM:ALL:TRACE 1

: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 or queries the transmission byte order for waveform data in word format, which is 2 bytes or longer in length.

Syntax :WAVEform:BYTeorder {LSBFirst|MSBFirst}
:WAVEform:BYTeorder?

Example :WAVEFORM:BYTEORDER LSBFIRST
:WAVEFORM:BYTEORDER?
-> :WAVEFORM:BYTEORDER LSBFIRST

:WAVEform:END

Function Sets or queries the end data point in the waveform specified by :WAVEform:TRACe.

Syntax :WAVEform:END {<NRf>}
:WAVEform:END?
<NRf> = 1 to 999999999

Example :WAVEFORM:END 12499
:WAVEFORM:END?
-> :WAVEFORM:END 12499

Description Use :WAVEform:LENGth? to query the total number of data points.

:WAVEform:FORMat

Function Sets or queries the format of the waveform data to be sent.

Syntax :WAVEform:FORMat {ASCIi|BYTE|RBYTe|WORD}
:WAVEform:FORMat?

Example :WAVEFORM:FORMAT ASCII
:WAVEFORM:FORMAT?
-> :WAVEFORM:FORMAT ASCII

Description For information about the differences depending on this format, see the description of the :WAVEform:SEND? command.

: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 used to convert the waveform data specified by :WAVEform:TRACe to physical values.

Syntax :WAVEform:OFFSet?

Example :WAVEFORM:OFFSET?
-> :WAVEFORM:OFFSET 0.000E+00

Description • This offset is used when converting the <block data> output by the :WAVEform:SEND? command into physical values.
• When :SYSTem:OCANcel is set to ON, the instrument returns 0.

:WAVEform:POSition?

Function Queries the vertical position used to convert waveform data into voltage when :WAVEform:FORMat is set to RBYTe.

Syntax :WAVEform:POSition?

Example :WAVEFORM:POSITION?
-> :WAVEFORM:POSITION 128

:WAVEform:RANGe?

Function Queries the range value used to convert the waveform data specified by :WAVEform:TRACe to physical values.

Syntax :WAVEform:RANGe?

Example :WAVEFORM:RANGE?
-> :WAVEFORM:RANGE 5.000E+00

Description This range is used when converting the <block data> output by the :WAVEform:SEND? command into physical values.

:WAVEform:RECOrd

Function Sets or queries the record number that WAVEform commands will be applied to.

Syntax :WAVEform:RECOrd {<NRf>|AVERage|
MINimum}
:WAVEform:RECOrd?

Example :WAVEFORM:RECORD 0
:WAVEFORM:RECORD?
-> :WAVEFORM:RECORD 0

Description If AVERage is specified, the command will be applied to history average waveform. Specify "MINimum" to specify the minimum record number. The settable record numbers vary depending on the acquisition settings. For details, see the Features Guide.

:WAVEform:RECOrd? MINimum

Function Queries the smallest record number in the source channel history.

Syntax :WAVEform:RECOrd? MINimum

Example :WAVEFORM:RECORD? MINIMUM
-> :WAVEFORM:RECORD -49999

:WAVEform:SEND?

Function Queries the waveform data specified by :WAVEform:TRACe.

Syntax :WAVEform:SEND? [<NRf>]
<NRf> = 1 to 100000
The <NRf> range varies depending on the record length setting.

Example :WAVEFORM:SEND?
-> #9(9-digit number of bytes)(data sequence) or <NRf>,<NRf>,...

Description • The :WAVEform:SEND? output format varies depending on the :WAVEform:FORMat setting.

- (1) When set to "ASCIi," the information is returned in this form: <Voltage>,<Voltage>,...<Voltage>
- (2) When set to "BYTE" or "WORD," the information is returned in <block data> format. The following equation can be used to make the conversion. Voltage (computed value) = (range×data÷Division*) + offset
 - * "BYTE": Division = 12.5
 - "WORD": Division = 3200
- (3) When set to "RBYTe," the information is returned in <block data> format. The following equation can be used to make the conversion. Voltage (computed value) = (range×(data – Position)÷Division*) + offset
 - * Division = 25
 - Position = The return value of " :WAVEform:POSition?"

• <NRf> can be omitted. If you specify <NRf>, waveform data is queried <NRf> times in order starting from the record located at the record number specified by :WAVEform:RECOrd – <NRf>.

:WAVEform:SIGN?

Function Queries whether signs are included in the binary data of the source waveform specified by :WAVEform:TRACe when the data is queried.

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

5.34 WAVEform Group

:WAVEform:START

Function Sets or queries the start data point in the waveform specified by :WAVEform:TRACe.

Syntax :WAVEform:START {<NRf>}
:WAVEform:START?
<NRf> = 0 to 999999999

Example :WAVEFORM:START 0
:WAVEFORM:START?
-> :WAVEFORM:START 0

:WAVEform:TRACe

Function Sets or queries the waveform that WAVEform commands will be applied to.

Syntax :WAVEform:TRACe {<NRf>|LOGic|
MATH<x>}
:WAVEform:TRACe?
<NRf> = 1 to 4
<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 beginning 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

5.35 WPARameter Group

:WPARameter<x>?

Function Queries all waveform parameter measurement settings.

Syntax :WPARameter<x>?
<x> = 1 to 2

:WPARameter<x>:DISPlay

Function Sets or queries the on/off state of waveform parameter measurement display.

Syntax :WPARameter<x>:DISPlay {<Boolean>}
:WPARameter<x>:DISPlay?
<x> = 1 to 2

Example :WPARAMETER1:DISPLAY ON
:WPARAMETER1:DISPLAY?
-> :WPARAMETER1:DISPLAY 1

:WPARameter<x>:HISTogram?

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

Syntax :WPARameter<x>:HISTogram?
<x> = 1 to 2

:WPARameter<x>:HISTogram:MEASure?

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

Syntax :WPARameter<x>:HISTogram:MEASure?
<x> = 1 to 2

:WPARameter<x>:HISTogram:MEASure:MODE

Function Sets or queries the auto measurement mode of the histogram display of waveform parameter measurement.

Syntax :WPARameter<x>:HISTogram:MEASure:
MODE {OFF|PARAMeter}
:WPARameter<x>:HISTogram:MEASure:
MODE?
<x> = 1 to 2

Example :WPARAMETER1:HISTOGRAM:MEASURE:
MODE OFF
:WPARAMETER1:HISTOGRAM:MEASURE:MODE?
-> :WPARAMETER1:HISTOGRAM:MEASURE:
MODE OFF

:WPARameter<x>:HISTogram:MEASure:PARa meter?

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

Syntax :WPARameter<x>:HISTogram:MEASure:
PARameter?
<x> = 1 to 2

:WPARameter<x>:HISTogram:MEASure:PARa meter:ALL

Function Collectively turns on or off all the histogram parameters of waveform parameter measurement.

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

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

:WPARameter<x>:HISTogram:MEASure:PARa meter:<Parameter>?

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

Syntax :WPARameter<x>:HISTogram:MEASure:
PARameter:<Parameter>?
<x> = 1 to 2
<Parameter> = {C1|C2|DC|MAXimum|MEAN|ME
Dian|MINimum|PEAK|SD2integ|SD3integ|SDEVi
ation|SDInteg}

:WPARameter<x>:HISTogram:MEASure:PARa meter:<Parameter>:STATE

Function Sets or queries the on/off state the histogram parameters of waveform parameter measurement display.

Syntax :WPARameter<x>:HISTogram:MEASure:
PARameter:<Parameter>:
STATE {<Boolean>}
:WPARameter<x>:HISTogram:MEASure:
PARameter:<Parameter>:STATE?
<x> = 1 to 2
<Parameter> = {C1|C2|DC|MAXimum|MEAN|ME
Dian|MINimum|PEAK|SD2integ|SD3integ|SDEVi
ation|SDInteg}

Example The following example is for the maximum value.
:WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER1:MAXIMUM:STATE ON
:WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER1:MAXIMUM:STATE?
-> :WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER1:MAXIMUM:STATE 1

5.35 WPARameter Group

:WPARameter<x>:HISTogram:MEASure:PARa meter:<Parameter>:VALue?

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

Syntax :WPARameter<x>:HISTogram:MEASure:
PARameter:<Parameter>:VALue?
<x> = 1 to 2
<Parameter> = {C1|C2|DC|MAXimum|MEAN|ME
Dian|MINimum|PEAK|SD2integ|SD3integ|SDEVI
ation|SDINteg}

Example The following example is for the maximum value.
:WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER:MAXIMUM:VALUE?
-> :WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER:MAXIMUM:VALUE 1.000E+00

:WPARameter<x>:HISTogram:MEASure:PARa meter:POSition<y>

Function Sets or queries the position a parameter of waveform parameter measurement histogram.

Syntax :WPARameter<x>:HISTogram:MEASure:
PARameter:POSition<y> {<NRF>}
:WPARameter<x>:HISTogram:MEASure:
PARameter:POSition<y>?
<x> = 1 to 2
<y> = 1 to 2
<NRF> = -5 to 5 (div)

Example :WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER1:POSITION 1
:WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER1:POSITION?
-> :WPARAMETER1:HISTOGRAM:MEASURE:
PARAMETER1:POSITION 1.000E+00

:WPARameter<x>:ITEM

Function Sets or queries a waveform parameter of waveform parameter measurement.

Syntax :WPARameter<x>:ITEM {<NRF>|BIT<y>|
CALC<y>|MATH<y>,<Parameter>[,2]}
:WPARameter<x>:ITEM?
WPARameter<x>'s <x> = 1 or 2
<NRF> = 1 to 4
BIT<y>'s <y> = 1 to 8
CALC<y>'s <y> = 1 to 4
MATH<y>'s <y> = 1 to 4
<Parameter> = {AMPLitude|AVERage|AVGFreq|
AVGPeriod|BWIDth|DELay|DT|DUTYcycle|ENUM
ber|FALL|FREQuency|I2T|HIGH|LOW|MAXimum
|MINimum|NOVershoot|NWDth|P|PABS|PERiod|
PN|PNUMBER|PON|POVershoot|PP|PTOFF|PTON
|PTOPeak|PTOTal|PWIDTH|RISE|RMS|SDEVIatio
n|TY1Integ|TY2Integ|V1|V2|WH|WHABs|WHN|W
HON|WHP|WHTOFF|WHTON|WHTOTal|Z}

Example :WPARAMETER1:ITEM 1,AVERAGE
:WPARAMETER1:ITEM?
-> :WPARAMETER1:ITEM 1,AVERAGE

Description • For BIT<y>, the available <Parameter> values are {AVGFreq|DELay|DUTYcycle|FREQuency|PERiod|PNUMBER}.

• For CALC<y>, the parameter is not necessary.

• For <NRF> and MATH<y>, the available <Parameter> values are {AMPLitude|AVERage|AVGFreq|AVGPeriod|BWIDth|DELay|DT|DUTYcycle|ENUMber|FALL|FREQuency|HIGH|LOW|MAXimum|MINimum|NOVershoot|NWDth|PERiod|PNUMBER|POVershoot|PTOPeak|PWIDTH|RISE|RMS|SDEVIation|TY1Integ|TY2Integ|V1|V2}.

• When <Parameter> is set to {PON|PTOFF|PTON|PTOTal|WHON|WHTOFF|WHTON|WHTOTal}, the [, 2] after the <Parameter> cannot be specified.

:WPARameter<x>:MODE

Function Sets or queries the waveform parameter measurement mode.

Syntax :WPARameter<x>:MODE {HISTogram|
TREND}
:WPARameter<x>:MODE?
<x> = 1 to 2

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

:WPARameter<x>:TREND?

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

Syntax :WPARameter<x>:TREND?
<x> = 1 to 2

:WPARameter<x>:TREND:AScale

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

Syntax :WPARameter<x>:TREND:AScale
<x> = 1 to 2

Example :WPARAMETER1:TREND:ASCALE

:WPARameter<x>:TREND:CUSor?

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

Syntax :WPARameter<x>:TREND:CUSor?
<x> = 1 to 2

:WPARameter<x>:TREND:CUSor:C<y>?

Function Queries the measured value of a cursor of waveform parameter measurement.

Syntax :WPARameter<x>:TREND:CUSor:C<y>?
<x> = 1 to 2
<y> = 1 to 2

Example :WPARAMETER1:TREND:CUSor:C1?
-> :WPARAMETER1:TREND:CUSor:
C1 1.000E+00

:WPARameter<x>:TREND:CUSor:DC?

Function Queries the measured value of a cursor of the trend of waveform parameter measurement.

Syntax :WPARameter<x>:TREND:CUSor:DC?
<x> = 1 to 2

Example :WPARAMETER1:TREND:CUSor:DC?
-> :WPARAMETER1:TREND:CUSor:
DC 1.000E+00

:WPARameter<x>:TREND:CUSor:MODE

Function Sets or queries the auto measurement mode of the trend of waveform parameter measurement.

Syntax :WPARameter<x>:TREND:CUSor:
MODE {<Boolean>}
:WPARameter<x>:TREND:CUSor:MODE?
<x> = 1 to 2

Example :WPARAMETER1:TREND:CUSor:MODE ON
:WPARAMETER1:TREND:CUSor:MODE?
-> :WPARAMETER1:TREND:CUSor:MODE 1

:WPARameter<x>:TREND:CUSor:POSition<y>

Function Sets or queries the position a cursor of the trend of waveform parameter measurement display.

Syntax :WPARameter<x>:TREND:CUSor:
POSition<y> {<Nrf>}
:WPARameter<x>:TREND:CUSor:
POSition<y>?
<x> = 1 to 2
<y> = 1 to 2
<Nrf> = -5 to 5 (div)

Example :WPARAMETER1:TREND:CUSor:
POSITION1 1
:WPARAMETER1:TREND:CUSor:POSITION1?
-> :WPARAMETER1:TREND:CUSor:
POSITION1 1.000E+00

:WPARameter<x>:TREND:HRANge

Function Sets or queries the trend display source window for waveform parameter measurement.

Syntax :WPARameter<x>:TREND:HRANge {MAIN|
Z1|Z2}
:WPARameter<x>:TREND:HRANge?
<x> = 1 to 2

Example :WPARAMETER1:TREND:HRANge MAIN
:WPARAMETER1:TREND:HRANge?
-> :WPARAMETER1:TREND:HRANge MAIN

:WPARameter<x>:TREND:HSPan

Function Sets or queries the trend display horizontal span for waveform parameter measurement.

Syntax :WPARameter<x>:TREND:HSPan {<Nrf>}
:WPARameter<x>:TREND:HSPan?
<x> = 1 to 2
<Nrf> = 1 to 100000

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

:WPARameter<x>:TREND:VERTical

Function Sets or queries the vertical range of the trend of waveform parameter measurement.

Syntax :WPARameter<x>:TREND:
VERTical {<Nrf>,<Nrf>}
:WPARameter<x>:TREND:VERTical?
<x> = 1 to 2

<Nrf> = -1.0000E+31 to 1.0000E+31
Example :WPARAMETER1:TREND:VERTICAL 1
:WPARAMETER1:TREND:VERTICAL?
-> :WPARAMETER1:TREND:
VERTICAL 1.00000E+00

5.35 WPARameter Group

:WPARameter<x>:VTDisplay

Function Sets or queries the VT waveform display on/off state.

Syntax :WPARameter<x>:VTDisplay {<Boolean>}
:WPARameter<x>:VTDisplay?
<x> = 1 to 2

Example :WPARAMETER1:VTDISPLAY ON
:WPARAMETER1:VTDISPLAY?
-> :WPARAMETER1:VTDISPLAY 1

:WPARameter<x>:WAIT?

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

Syntax :WPARameter<x>:WAIT? {<NRf>}
<x> = 1 to 2
<NRf> = 1 to 36000 (timeout value, 10 ms)

Example :WPARAMETER1:WAIT? 10
-> :WPARAMETER1:WAIT 1

Description If the execution of automated measurement is completed within the timeout period, 0 is returned. If it is not completed or if automated measurement is not being performed, 1 is returned. Even if you set a long timeout period, 0 is returned as soon as the automated measurement is completed.

5.36 XY Group

:XY<x>?

Function Queries all XY display settings.

Syntax :XY<x>?
<x> = 1 to 2

:XY<x>:DISPlay

Function Sets or queries whether to show (ON) or hide (OFF) the XY display on the screen.

Syntax :XY<x>:DISPlay {<Boolean>}
:XY<x>:DISPlay?
<x> = 1 to 2

Example :XY1:DISPlay ON
:XY1:DISPlay?
-> :XY1:DISPlay 1

:XY<x>:MEASure?

Function Queries all automated measurement settings of XY display.

Syntax :XY<x>:MEASure?
<x> = 1 to 2

:XY<x>:MEASure:CURSor?

Function Queries all cursor measurement settings of XY display.

Syntax :XY<x>:MEASure:CURSor?
<x> = 1 to 2

:XY<x>:MEASure:CURSor:DX?

Function Queries all settings for the voltage difference between the XY display's horizontal cursors.

Syntax :XY<x>:MEASure:CURSor:DX?
<x> = 1 to 2

:XY<x>:MEASure:CURSor:DX:STATe

Function Sets or queries the on/off status of the voltage difference between the XY display's horizontal cursors.

Syntax :XY<x>:MEASure:CURSor:DX:
STATe {<Boolean>}
:XY<x>:MEASure:CURSor:DX:STATe?
<x> = 1 to 2

Example :XY1:MEASURE:CURSOR:DX:STATE ON
:XY1:MEASURE:CURSOR:DX:STATE?
-> :XY1:MEASURE:CURSOR:DX:STATE 1

:XY<x>:MEASure:CURSor:DX:VALue?

Function Queries the voltage difference between the XY display's horizontal cursors.

Syntax :XY<x>:MEASure:CURSor:DX:VALue?
<x> = 1 to 2

Example :XY1:MEASURE:CURSOR:DX:VALUE?
-> :XY1:MEASURE:CURSOR:DX:
VALUE 6.000E+00

:XY<x>:MEASure:CURSor:DY?

Function Queries all settings for the voltage difference between the XY display's vertical cursors.

Syntax :XY<x>:MEASure:CURSor:DY?
<x> = 1 to 2

:XY<x>:MEASure:CURSor:DY:STATe

Function Sets or queries the on/off status of the voltage difference between the XY display's vertical cursors.

Syntax :XY<x>:MEASure:CURSor:DY:
STATe {<Boolean>}
:XY<x>:MEASure:CURSor:DY:STATe?
<x> = 1 to 2

Example :XY1:MEASURE:CURSOR:DY:STATE ON
:XY1:MEASURE:CURSOR:DY:STATE?
-> :XY1:MEASURE:CURSOR:DY:STATE 1

:XY<x>:MEASure:CURSor:DY:VALue?

Function Queries the voltage difference between the XY display's vertical cursors.

Syntax :XY<x>:MEASure:CURSor:DY:VALue?
<x> = 1 to 2

Example :XY1:MEASURE:CURSOR:DY:VALUE?
-> :XY1:MEASURE:CURSOR:DY:
VALUE 6.000E+00

:XY<x>:MEASure:CURSor:X<y>?

Function Queries all horizontal cursor settings of XY display.

Syntax :XY<x>:MEASure:CURSor:X<y>?
<x> = 1 to 2
<y> = 1 to 2

5.36 XY Group

:XY<x>:MEASure:CURSor:X<y>:POSition

Function Sets or queries a horizontal cursor position on the XY display.

Syntax :XY<x>:MEASure:CURSor:X<y>:
POSition {<NRf>}
:XY<x>:MEASure:CURSor:X<y>:POSition?
<x> = 1 to 2
<y> = 1 to 2
<NRf> = -4 to 4 (div)

Example :XY1:MEASURE:CURSOR:X1:POSITION 1
:XY1:MEASURE:CURSOR:X1:POSITION?
-> :XY1:MEASURE:CURSOR:X1:
POSITION 1.000E+00

:XY<x>:MEASure:CURSor:X<y>:STATe

Function Sets or queries the on/off status of the voltage of an XY display's horizontal cursor.

Syntax :XY<x>:MEASure:CURSor:X<y>:
STATe {<Boolean>}
:XY<x>:MEASure:CURSor:X<y>:STATe?
<x> = 1 to 2
<y> = 1 to 2

Example :XY1:MEASURE:CURSOR:X1:STATE ON
:XY1:MEASURE:CURSOR:X1:STATE?
-> :XY1:MEASURE:CURSOR:X1:STATE 1

:XY<x>:MEASure:CURSor:X<y>:VALue?

Function Queries the voltage at a XY display's horizontal cursor.

Syntax :XY<x>:MEASure:CURSor:X<y>:VALue?
<x> = 1 to 2
<y> = 1 to 2

Example :XY1:MEASURE:CURSOR:X1:VALUE?
-> :XY1:MEASURE:CURSOR:X1:
VALUE 1.000E+00

:XY<x>:MEASure:CURSor:Y<y>?

Function Queries all vertical cursor settings of XY display.

Syntax :XY<x>:MEASure:CURSor:Y<y>?
<x> = 1 to 2
<y> = 1 to 2

:XY<x>:MEASure:CURSor:Y<y>:POSition

Function Sets or queries a vertical cursor position on the XY display.

Syntax :XY<x>:MEASure:CURSor:Y<y>:
POSition {<NRf>}
:XY<x>:MEASure:CURSor:Y<y>:POSition?
<x> = 1 to 2
<y> = 1 to 2
<NRf> = -4 to 4 (div)

Example :XY1:MEASURE:CURSOR:Y1:POSITION 1
:XY1:MEASURE:CURSOR:Y1:POSITION?
-> :XY1:MEASURE:CURSOR:Y1:
POSITION 1.000E+00

:XY<x>:MEASure:CURSor:Y<y>:STATe

Function Sets or queries the on/off status of the voltage of an XY display's vertical cursor.

Syntax :XY<x>:MEASure:CURSor:Y<y>:
STATe {<Boolean>}
:XY<x>:MEASure:CURSor:Y<y>:STATe?
<x> = 1 to 2
<y> = 1 to 2

Example :XY1:MEASURE:CURSOR:Y1:STATE ON
:XY1:MEASURE:CURSOR:Y1:STATE?
-> :XY1:MEASURE:CURSOR:Y1:STATE 1

:XY<x>:MEASure:CURSor:Y<y>:VALue?

Function Queries the voltage at a XY display's vertical cursor.

Syntax :XY<x>:MEASure:CURSor:Y<y>:VALue?
<x> = 1 to 2
<y> = 1 to 2

Example :XY1:MEASURE:CURSOR:Y1:VALUE?
-> :XY1:MEASURE:CURSOR:Y1:
VALUE 1.000E+00

:XY<x>:MEASure:INTeg?

Function Queries all integral settings of XY display.

Syntax :XY<x>:MEASure:INTeg?
<x> = 1 to 2

:XY<x>:MEASure:INTeg:LOOP

Function Sets or queries the integral method on the XY display.

Syntax :XY<x>:MEASure:INTeg:LOOP {CLOSe|
OPEN}
:XY<x>:MEASure:INTeg:LOOP?
<x> = 1 to 2

Example :XY1:MEASURE:INTEG:LOOP CLOSE
:XY1:MEASURE:INTEG:LOOP?
-> :XY1:MEASURE:INTEG:LOOP CLOSE

:XY<x>:MEASure:INTEg:POLarity

Function Sets or queries the integral direction on the XY display.

Syntax :XY<x>:MEASure:INTEg:POLarity {CCW|CW}
:XY<x>:MEASure:INTEg:POLarity?
<x> = 1 to 2

Example :XY1:MEASURE:INTEG:POLARITY CCW
:XY1:MEASURE:INTEG:POLARITY?
-> :XY1:MEASURE:INTEG:POLARITY CCW

:XY<x>:MEASure:INTEg:VALue?

Function Queries the integral value on the XY display.

Syntax :XY<x>:MEASure:INTEg:VALue?
<x> = 1 to 2

Example :XY1:MEASURE:INTEG:VALUE?
-> :XY1:MEASURE:INTEG:
VALUE 1.000E+00

:XY<x>:MEASure:MODE

Function Sets or queries the automated measurement mode on the XY display.

Syntax :XY<x>:MEASure:MODE {CURSor|INTEg|OFF}
:XY<x>:MEASure:MODE?
<x> = 1 to 2

Example :XY1:MEASURE:MODE CURSOR
:XY1:MEASURE:MODE?
-> :XY1:MEASURE:MODE CURSOR

:XY<x>:RANGe

Function Sets or queries the VT waveform source window for the XY display.

Syntax :XY<x>:RANGe{MAIN|Z1|Z2}
:XY<x>:RANGe?
<x> = 1 to 2

Example :XY1:RANGE MAIN
:XY1:RANGE?
-> :XY1:RANGE MAIN

:XY<x>:SPLit

Function Sets or queries whether the XY1/XY2 display screen is split on the XY display.

Syntax :XY<x>:SPLit {<Boolean>}
:XY<x>:SPLit?
<x> = 1 to 2

Example :XY1:SPLIT ON
:XY1:SPLIT?
-> :XY1:SPLIT 1

:XY<x>:TRANGe (Time Range)

Function Sets or queries the VT waveform range for the XY display.

Syntax :XY<x>:TRANGe {<NRf>,<NRf>}
:XY<x>:TRANGe?
<x> = 1 to 2
<NRf>, <NRf> = -5 to 5 (div), (10 div/display record length) steps

Example :XY1:TRANGE -4,4
:XY1:TRANGE?
-> :XY1:TRANGE -4.00,4.00

:XY<x>:VTDisPlay

Function Sets or queries the on/off state of the VT waveform display on the XY display.

Syntax :XY<x>:VTDisPlay {<Boolean>}
:XY<x>:VTDisPlay?
<x> = 1 to 2

Example :XY1:VTDISPLAY ON
:XY1:VTDISPLAY?
-> :XY1:VTDISPLAY 1

:XY<x>:XTRace

Function Sets or queries the channel that is assigned to the X-axis on the XY display.

Syntax :XY<x>:XTRace {<NRf>|MATH<y>}
:XY<x>:XTRace?
<x> = 1 to 2
<NRf> = 1 to 4
MATH<y>'s <y> = 1 to 4

Example :XY1:XTRACE 1
:XY1:XTRACE?
-> :XY1:XTRACE 1

:XY<x>:YTRace

Function Sets or queries the channel that is assigned to the Y-axis on the XY display.

Syntax :XY<x>:YTRace {<NRf>|MATH<y>}
:XY<x>:YTRace?
<x> = 1 to 2
<NRf> = 1 to 4
MATH<y>'s <y> = 1 to 4

Example :XY1:YTRACE 1
:XY1:YTRACE?
-> :XY1:YTRACE 1

5.37 ZOOM Group

:ZOOM<x>?

Function Queries all waveform zoom settings.

Syntax :ZOOM<x>?
<x> = 1 to 2

:ZOOM<x>:ALLOCATION?

Function Queries all zoom source waveform settings.

Syntax :ZOOM<x>:ALLOCATION?
<x> = 1 to 2

:ZOOM<x>:ALLOCATION:{CHANNEL<y>|MATH<y>}

Function Sets or queries the zoom source waveform.

Syntax :ZOOM<x>:ALLOCATION:{CHANNEL<y>|
MATH<y>} {<Boolean>}
:ZOOM<x>:ALLOCATION:{CHANNEL<y>|
MATH<y>}?
<x> = 1 to 2
<y> = 1 to 4

Example :ZOOM1:ALLOCATION:CHANNEL1 ON
:ZOOM1:ALLOCATION:CHANNEL1?
-> :ZOOM1:ALLOCATION:CHANNEL1 1

:ZOOM<x>:DISPLAY

Function Sets or queries the zoom waveform display on/off state.

Syntax :ZOOM<x>:DISPLAY {<Boolean>}
:ZOOM<x>:DISPLAY?
<x> = 1 to 2

Example :ZOOM1:DISPLAY ON
:ZOOM1:DISPLAY?
-> :ZOOM1:DISPLAY 1

:ZOOM<x>:FORMAT

Function Sets or queries the zoom waveform display format.

Syntax :ZOOM<x>:FORMAT {DUAL|HEXA|MAIN|
OCTal|QUAD|SINGLE|TRIad}
:ZOOM<x>:FORMAT?
<x> = 1 to 2

Example :ZOOM1:FORMAT DUAL
:ZOOM1:FORMAT?
-> :ZOOM1:FORMAT DUAL

:ZOOM<x>:MAG

Function Sets or queries the magnification of a zoom waveform display window.

Syntax :ZOOM<x>:MAG {<Nrf>}
:ZOOM<x>:MAG?
<x> = 1 to 2
<Nrf> = See the Features Guide for this information.

Example :ZOOM1:MAG 2.5
:ZOOM1:MAG?
-> :ZOOM1:MAG 2.5

:ZOOM<x>:MAGFine

Function Sets or queries the magnification (fine) of a zoom waveform display window.

Syntax :ZOOM<x>:MAGFine {<Nrf>}
:ZOOM<x>:MAGFine?
<x> = 1 to 2
<Nrf> = See the Features Guide for this information.

Example :ZOOM1:MAGFINE 2.5
:ZOOM1:MAGFINE?
-> :ZOOM1:MAGFINE 2.5

:ZOOM<x>:MAIN

Function Sets or queries the zoom waveform display ratio.

Syntax :ZOOM<x>:MAIN {20|50|OFF}
:ZOOM<x>:MAIN?
<x> = 1 to 2

Example :ZOOM1:MAIN 20
:ZOOM1:MAIN?
-> :ZOOM1:MAIN 20

:ZOOM<x>:POSITION

Function Sets or queries the position of a zoom box.

Syntax :ZOOM<x>:POSITION {<Nrf>}
:ZOOM<x>:POSITION?
<x> = 1 to 2
<Nrf> = -5 to 5 (div), 10 div/display record length steps

Example :ZOOM1:POSITION 2
:ZOOM1:POSITION?
-> :ZOOM1:POSITION 2.00E+00

:ZOOM<x>:VERTical?

Function Queries all vertical zoom settings.

Syntax :ZOOM<x>:VERTical?
<x> = 1 to 2

:ZOOM<x>:VERTical:{CHANnel<y>|MATH<y>}?<y>}

Function Queries all settings related to a vertical zoom trace.

Syntax :ZOOM<x>:VERTical:{CHANnel<y>|
MATH<y>}?
<x> = 1 to 2
<y> = 1 to 4

**:ZOOM<x>:VERTical:{CHANnel<y>|MATH<y>}
:MAG**

Function Sets or queries the vertical zoom factor.

Syntax :ZOOM<x>:VERTical:{CHANnel<y>|
MATH<y>}:MAG {<NRf>}
:ZOOM<x>:VERTical:{CHANnel<y>|
MATH<y>}:MAG?
<x> = 1 to 2
<y> = 1 to 4
<NRf> = See the Features Guide for this
information.

Example :ZOOM1:VERTICAL:CHANNEL1:MAG 2.5
:ZOOM1:VERTICAL:CHANNEL1:MAG?
-> :ZOOM1:VERTICAL:CHANNEL1:MAG 2.5

**:ZOOM<x>:VERTical:{CHANnel<y>|MATH<y>}
:POSition**

Function Sets or queries the vertical zoom position.

Syntax :ZOOM<x>:VERTical:{CHANnel<y>|
MATH<y>}:POSition {<NRf>}
:ZOOM<x>:VERTical:{CHANnel<y>|
MATH<y>}:POSition?
<x> = 1 to 2
<y> = 1 to 4
<NRf> = -4 to 4 (div)

Example :ZOOM1:VERTICAL:CHANNEL1:POSITION 2
:ZOOM1:VERTICAL:CHANNEL1:POSITION?
-> :ZOOM1:VERTICAL:CHANNEL1:
POSITION 2.00E+00

:ZOOM<x>:VERTical:RESet

Function Resets all vertical zoom settings.

Syntax :ZOOM<x>:VERTical:RESet
<x> = 1 to 2

Example :ZOOM1:VERTICAL:RESET

:ZOOM<x>:VERTical:TRACe

Function Sets or queries the trace to show on the vertical zoom window.

Syntax :ZOOM<x>:VERTical:TRACe {<NRf>|
MATH<y>}
:ZOOM<x>:VERTical:TRACe?
<x> = 1 to 2
<NRf> = 1 to 4
MATH<y>'s <y> = 1 to 4

Example :ZOOM1:VERTICAL:TRACE 1
:ZOOM1:VERTICAL:TRACE?
-> :ZOOM1:VERTICAL:TRACE 1

5.38 Common Command Group

The commands in this group are defined in USBTMC-USB488 and are independent from the instrument-specific functions. There are no front panel keys that correspond to the commands in this group.

***CAL? (CALibrate)**

Function Executes calibration and queries the result.

Syntax *CAL?

Example *CAL?
-> 0

Description If the calibration ends 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 information about each register and queue, see chapter 6.

***ESE (standard Event Status Enable register)**

Function Sets or queries the standard event enable register.

Syntax *ESE {<NRf>}
*ESE?
<NRf> = 0 to 255

Example *ESE 251
*ESE?
-> 251

Description

- Specify the value as a sum of the values of each bit in decimal format.
- 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. This 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 information about the standard event enable register, see section 6.3.

***ESR? (standard Event Status Register)**

Function Queries and clears the standard event register.

Syntax *ESR?

Example *ESR?
-> 32

Description

- A sum of the values of each bit is returned in decimal format.
- When an SRQ is sent, you can check what types of events have occurred.
- For example, if a value of 32 is returned, this indicates that the standard event register is set to 00100000. This means that the SRQ occurred due to a command syntax error.
- A query using *ESR? will clear the contents of the Standard Event Register.
- For information about the standard event enable register, see section 6.3.

***IDN? (IDeNtify)**

Function Queries the instrument model.

Syntax *IDN?

Example *IDN?
-> YOKOGAWA,710120,08ELP1024,F1.01

Description This command returns a string in the following format: <Manufacturer>, <Model>, <Serial number>, <Firmware version>. For <Model>, the instrument returns "DLM3034HD" and "DLM3054HD" for DLM3034HD and DLM3054HD, respectively.

***OPC (OPeration Complete)**

Function Sets bit 0 (the OPC bit) of the standard event register to 1 upon the completion of the specified overlap command.

Syntax *OPC

Example *OPC

Description

- For information about how to synchronize a program using *OPC, see page 4-12.
- Use :COMMunicate:OPSE to specify the overlap command.
- If *OPC is not the last command of the message, its operation is not guaranteed.

***OPC? (Operation Complete)**

Function When you send *OPC?, the instrument returns ASCII code 1 when the specified overlap command is completed.

Syntax *OPC?

Example *OPC?
-> 1

Description

- For information about how to synchronize a program using *OPC?, see page 4-12.
- Use :COMMunicate:OPSE to specify the overlap command.
- If *OPC? is not the last command of the message, its operation is not guaranteed.

***OPT? (OPTION)**

Function Queries the installed options.

Syntax *OPT?

Example *OPT?
-> 500MPOINTS,LOGIC,PROBPOWER4,
GPIB,STORAGE,USERDEFINE,PANALYZE,
UART,I2C,SPI,CAN,CAN FD,LIN,FLEXRAY,
SENT,CXPI

Description

- The instrument returns the availabilities of <memory model>, <switch logic>, <printer>, <rear-panel probe power>, <GP-IB>, <internal storage>, <user-defined computation>, <power supply analysis feature>, <UART analysis feature>, <I2C analysis feature>, <SPI analysis feature>, <CAN analysis feature>, <CAN FD analysis feature>, <LIN analysis feature>, <FlexRay analysis feature>, <CXPI analysis feature>, and so on.
- The *OPT? query must be the last query of a program message. An error occurs if there is a query after this query.

***RST (ReSeT)**

Function Initializes the settings.

Syntax *RST

Example *RST

Description Also clears *OPC and *OPC? commands that have been sent.

***SRE (Service Request Enable register)**

Function Sets or queries the service request enable register value.

Syntax *SRE <NRf>

*SRE?

<NRf> = 0 to 255

Example *SRE 239

*SRE?

-> 175

Description

- Specify the value as a sum of the values of each bit in decimal format.
- For example, specifying *SRE 239 will cause the standard enable register to be set to 11101111. In this case, bit 4 of the service request enable register is disabled. This 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 is therefore 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 information about the service request enable register, see page 6-2.

***STB? (STatus Byte)**

Function Queries the status byte register value.

Syntax *STB?

Example *STB?

-> 4

Description

- A sum of the values of each bit is returned as a decimal value.
- Because the register is read without executing serial polling, bit 6 is an MSS bit, not an RQS bit.
- For example, if a value of 4 is returned, this indicates that the status byte register is set to 00000100. This means that the error queue is not empty (in other words, an error occurred).
- A query using *STB? will not clear the contents of the status byte register.
- For information about the status byte register, see page 6-2.

5.38 Common Command Group

***TST?**

Function Executes a self-test and queries the result. The self-test consists of tests of each kind of internal memory.

Syntax *TST?

Example *TST?
-> 0

Description This command returns 0 if the self-test is successful and 1 if it is not.

***WAI (WAIt)**

Function Holds the execution of the subsequent command until the specified overlap command is completed.

Syntax *WAI

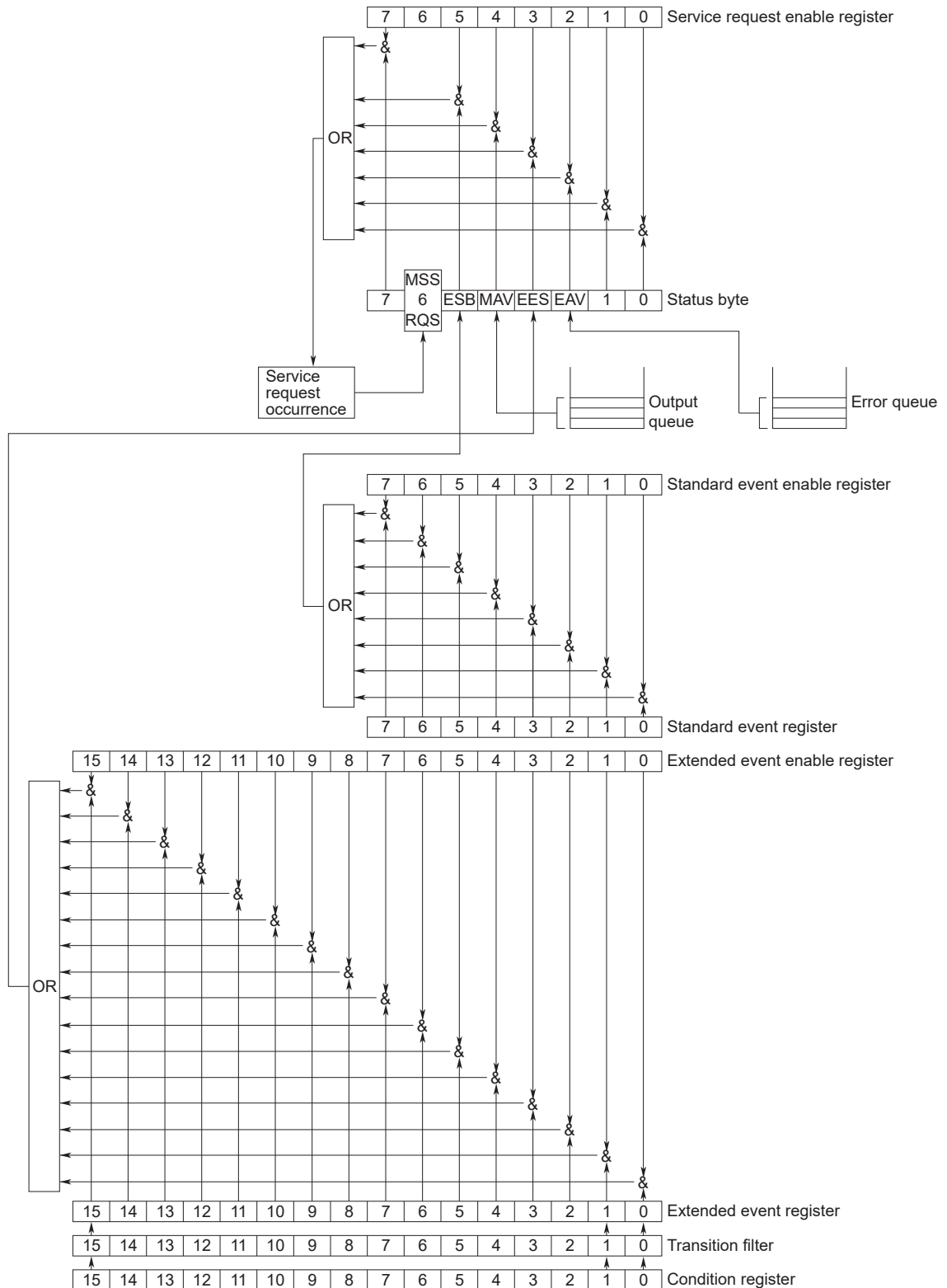
Example *WAI

Description • For information about how to synchronize a program using *WAI, see page 4-11.
• Use :COMMunicate:OPSE to specify the overlap command.

6.1 About Status Reports

Status reports

The figure below shows the format of status reports that are read by serial polling. This status report format is an extended version of the status report format defined in IEEE 488.2-1992.



Overview of registers and queues

Name	Function	Write	Read
Status byte	–	–	Serial polling (RQS), *STB? (MSS)
Service request enable register	Status byte mask	*SRE	*SRE?
Standard event register	Indicates device status changes	–	*ESR?
Standard event enable register	Standard event register mask	*ESE	*ESE?
Extended event register	Indicates device status changes	–	STATUS:EESR?
Extended event enable register	Extended event register mask	STATUS:EESE	STATUS:EESE?
Condition register	Current device status	–	STATUS:CONDition?
Transition filter	Conditions that change the extended event register	STATUS:FILTer<x>	STATUS:FILTer<x>?
Output queue	Stores response messages for queries	Query commands	
Error queue	Stores error numbers and messages	–	STATUS:ERRor?

Registers and queues that affect the status byte

The following registers affect the status byte bits.

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

The following registers are used to mask a bit so that the bit will not affect the status byte even when it is set to 1.

Service request enable register	Masks bits of the status byte.
Standard event register	Masks bits in the standard event register.
Extended event register	Masks bits in the extended event register.

Reading and writing to registers

For example, use the *ESE command to set the standard event enable register bits to 1 and 0. You can use the *ESE? command to query whether the standard event enable register bits are ones or zeros. For details on these commands, see chapter 5.

6.2 Status Byte

Status byte

	RQS						
7	6	ESB	MAV	EES	EAV	1	0
	MSS						

Bits 0, 1, and 7

Not used (always 0)

Bit 2 EAV (Error Available)

This bit is 1 when the error queue is not empty. In other words, this bit is set to 1 when an error occurs. See section 6.5.

Bit 3 EES (Extend Event Summary Bit)

This bit is 1 when the logical AND of the extended event register and its corresponding event register is 1. In other words, this bit is set to 1 when a certain event takes place inside the instrument. See section 6.4.

Bit 4 MAV (Message Available)

This bit is 1 when the output queue is not empty. In other words, this bit is set to 1 when there is data to be transmitted in response to a query. See section 6.5.

Bit 5 ESB (Event Summary Bit)

This bit is 1 when the logical AND of the standard event register and its corresponding event register is 1. In other words, this bit is set to 1 when a certain event takes place inside the instrument. For details, see page 6.3-6.

Bit 6 RQS (Request Service)/MSS (Master Status Summary)

This bit is 1 when the logical AND of the status byte excluding bit 6 and the service request enable register is 1. In other words, this bit is set to 1 when the instrument requests service from the controller.

RQS is set to 1 when the MSS bit changes from 0 to 1 and is 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. Do this 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.

Status byte operation

A service request is issued when bit 6 in the status byte becomes 1. Bit 6 is set to 1 when any other bit becomes 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 OR of a standard event register bit and its corresponding enable register bit is 1, then bit 5 (ESB) is set to 1. At this point, if bit 5 of the service request enable register is 1, bit 6 (MSS) is set to 1, and the instrument requests service from the controller.

You can check what type of event occurred by reading the contents of the status byte.

Reading the status byte

There are two ways to read the contents of the status byte.

***STB? query**

Bit 6 functions as MSS when a query is made using `*STB?`. This causes the MSS to be read. This query does not cause any of the status byte bits to be cleared after the status byte is read.

Serial polling

Serial polling causes bit 6 to function as an RQS bit. This causes the RQS to be read. After the status byte is read, only the RQS bit is cleared. You cannot read the MSS bit when serial polling is used.

Clearing the status byte

There is no way to clear all the bits in the status byte. The bits that are cleared for each operation are shown below.

***STB? query**

None of the bits are cleared.

Serial polling

Only the RQS bit is cleared.

When a *CLS command is received

When a `*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 status byte bits are cleared. Because the output queue is not cleared with a `*CLS` command, bit 4 (MAV) in the status byte is not affected. However, the output queue will 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)

This bit is 1 when the instrument is turned on.

Bit 6 URQ (User Request)

Not used (always 0)

Bit 5 CME (Command Error)

This bit is set to 1 when there is a command syntax error.

Example Incorrectly spelled command name; 9 used in octal data.

Bit 4 EXE (Execution Error)

This bit is set to 1 when the command syntax is correct, but the command cannot be executed in the current state.

Example The instrument receives a command whose parameter is outside the selectable range. An attempt is made to print a hard copy while the instrument is running.

Bit 3 DDE (Device Error)

This bit is set to 1 when a command cannot be executed for instrument's internal reasons other than a command syntax error or command execution error.

Bit 2 QYE (Query Error)

This bit is set to 1 when a query command is received, but the output queue is empty or the data is lost.

Example There is no response data, or 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)

This bit is set to 1 upon the completion of the operation designated by the *OPC command (see chapter 5).

Bit masking

To mask a certain bit of the standard event register so that it does not cause bit 5 (ESB) in the status byte to change, set the corresponding bit of the standard event enable register to 0.

For example, to mask bit 2 (QYE) so that ESB will not be set to 1 even if a query error occurs, set bit 2 of the standard event enable register to 0. Do this using the *ESE command. To query whether each bit of the standard event enable register is 1 or 0, use *ESE?. For details on the *ESE command, see chapter 5.

Standard event register operation

The standard event register indicates eight types of events that occur inside the instrument. When one of the bits in this register becomes 1 (and the corresponding bit of the standard event enable register is also 1), bit 5 (ESB) in the status byte is set to 1.

Example

1. A query error occurs.
2. Bit 2 (QYE) is set to 1.
3. When bit 2 of the standard event enable register is 1, bit 5 (ESB) in the status byte is set to 1.

You can also check what type of event occurred in the instrument by reading the contents of the standard event register.

Reading the standard event register

You can use the `*ESR?` command to read the contents of the standard event register. The register is cleared after it is read.

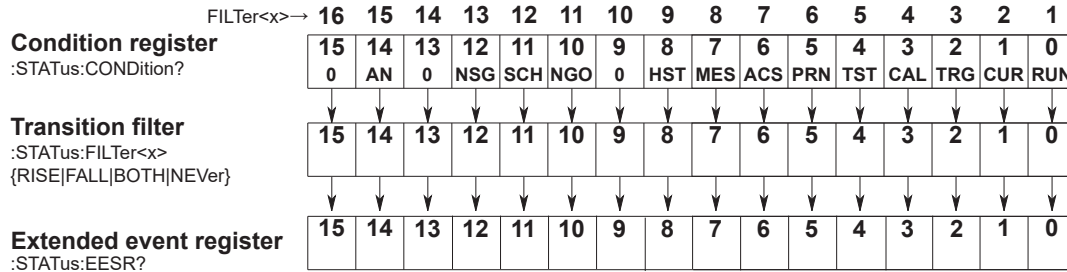
Clearing the standard event register

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 restarted.

6.4 Extended Event Register

The extended event register receives information about changes in the condition register, which indicates the instrument's internal condition. The information is the result of edge detection performed by the transition filter.



The condition register bits are described below.

Bit 0	RUN(Running)	This bit is 1 when waveform acquisition is in progress.
Bit 1	CUR(Cursor)	This bit is 1 when cursor measurement is in progress.
Bit 2	TRG(Awaiting trigger)	This bit is 1 when the instrument is waiting for a trigger.
Bit 3	CAL(Calibration)	This bit is 1 when calibration is in progress.
Bit 4	TST(Testing)	This bit is 1 when a self-test is in progress.
Bit 5	PRN(Printing)	This bit is 1 when the built-in printer is in operation, when data is being transmitted to an external USB or network printer, or when screen capture data is being saved.
Bit 6	ACS(Accessing)	This bit is 1 when a drive is being accessed.
Bit 7	MES(Measuring)	This bit is 1 when automated measurement of waveform parameters is in progress.
Bit 8	HST(History Search)	This bit is 1 when a history search is in progress.
Bit 10	NGO(Go/No-go)	This bit is 1 when a GO/NO-GO search is in progress.
Bit 11	SCH(Search)	This bit is set to 1 when a waveform search is in progress (this includes when a serial bus signal search is in progress).
Bit 12	NSG(N-Single)	This bit is 1 when consecutive acquisition is in progress when the trigger mode is set to Single (N).
Bit 14	AN(Analysis)	This bit is 1 when an analysis is in progress (this includes when a serial bus signal analysis is in progress).

The transition filter parameters detect changes in the specified condition register bits (numeric suffixes 1 to 16) and overwrite the extended event register in the following ways.

RISE	The specified extended event register bit is set to 0 when the corresponding condition register bit changes from 1 to 1.
FALL	The specified extended event register bit is set to 1 when the corresponding condition register bit changes from 1 to 0.
BOTH	The specified extended event register bit is set to 1 when the corresponding condition register bit changes from 0 to 1 or from 1 to 0.
NEVer	Always zero.

6.5 Output and Error Queues

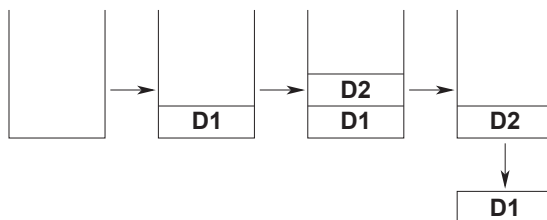
Output queue

The output queue stores query response messages. For example, if you send a `:WAVEform:SEND?` command, which requests for the transmission of acquired waveform data, the data is stored in the output queue until it is read.

As shown below, data is stored in order and read from the oldest message first. The output queue is cleared in the following cases.

- When a new message is received from the controller.
- When a deadlock occurs (see page 4-3).
- When a device clear command (`DCL` or `SDC`) is received.
- When the instrument is restarted.

The `*CLS` command does not clear the output queue. You can determine whether the output queue is empty by checking bit 4 (MAV) in the status byte.



Error queue

When an error occurs, the error queue stores the error number and message. For example, if the instrument receives an incorrect program message from the controller, the error number (113) and the error message ("Undefined header") are stored in the error queue when the instrument displays the error message.

You can use the `:STATUS:ERROR?` query to read the contents of the error queue. Like the output queue, the messages in the error queue are read from the oldest one first.

If the error queue overflows, the last message is replaced with the following message: 350, "Queue overflow."

The error queue is cleared in the following cases.

- When a `*CLS` command is received.
- When the instrument is restarted.

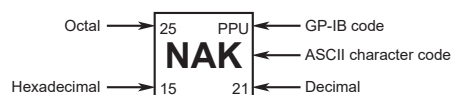
You can determine whether or not the error queue is empty by checking bit 2 (EAV) in the status byte.

Appendix 1 ASCII Character Codes

This section provides a table of ASCII character codes.

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

Example



Appendix 2 Error Messages

This section explains communication error messages.

- Messages can be displayed in English or Japanese on the instrument. However, when they are read from a PC or other similar device, messages are displayed in English.
- If servicing is necessary to solve the problem indicated by a message, contact your nearest YOKOGAWA dealer.
- Only communication error messages are listed here. For other error messages, see the User's Manual, IM DLM3054HD-03EN.
 - Communication syntax errors 100 to 199
 - Communication execution errors 200 to 299
 - Device-specific and other errors 300 to 399
 - Communication query errors 400 to 499
 - System errors (communication) 399

Listed below

Communication Syntax Errors (100 to 199)

Code	Message	Corrective Action	Page
100	Command error.	Check the spelling of the command.	Chapter 5
102	Syntax error.	A syntax error not covered by error codes 100 to 199.	Chapters 4 and 5
103	Invalid separator.	Separate data values with a comma.	4-1
104	Data type error.	Use a correct data type.	4-8 to 4-10
105	GET not allowed.	GET is not supported as a response to an interface message.	3-7
108	Parameter not allowed.	Check the number of data values.	4-8 and chapter 5
109	Missing parameter.	Be sure to include all necessary data values.	4-8 and chapter 5
111	Header separator error.	Use a comma to separate each header from its data.	4-1
112	Program mnemonic too long.	Check the command length.	Chapter 5
113	Undefined header.	Check the header.	4-6 and chapter 5
114	Header suffix out of range.	Check the header.	4-6 and chapter 5
120	Numeric data error.	A value must be specified where the syntax contains <NRf>.	4-8
123	Exponent too large.	Where the syntax contains <NR3>, make the exponent that follows E smaller.	4-8 and chapter 5
124	Too many digits.	Limit numeric values to 255 digits or less.	4-8 and chapter 5
128	Numeric data not allowed.	Use a data type other than <NRf>.	4-8 and chapter 5
131	Invalid suffix.	Check the units where the syntax contains <Voltage>, <Time>, <Frequency>, or <Current>.	4-9
134	Suffix too long.	Check the units where the syntax contains <Voltage>, <Time>, <Frequency>, or <Current>.	4-9
138	Suffix not allowed.	Units of measurement can only be used where the syntax contains <Voltage>, <Time>, <Frequency>, or <Current>.	4-9
141	Invalid character data.	Be sure to select one of the listed choices when the syntax contains {...}.	4-10 and chapter 5

Code	Message	Corrective Action	Page
144	Character data too long.	Check the spelling of the strings when the syntax contains {...}.	4-10 and chapter 5
148	Character data not allowed.	Use a data type other than {...}.	4-8 and chapter 5
150	String data error.	Enclose parameters with single or double quotation marks where the syntax contains <String>.	4-10
151	Invalid string data.	The <String> is either too long, or it contains an unusable character.	4-10 and chapter 5
158	String data not allowed.	Use a data type other than <String>.	4-8 and chapter 5
161	Invalid block data.	<Block data> cannot be used.	4-10 and chapter 5
168	Block data not allowed.	<Block data> cannot be used.	4-10 and chapter 5
171	Missing Right.	Mathematical operations cannot be used.	—
172	Invalid expression.	Mathematical operations cannot be used.	Chapter 5
178	Expression data not allowed.	Mathematical operations cannot be used.	Chapter 5
181	Invalid outside macro definition.	The instrument does not support the IEEE 488.2 macro specifications.	—

Communication Execution Errors (200 to 299)

Code	Message	Corrective Action	Page
221	Setting conflict.	Check settings that are related to each other.	Chapter 5
222	Data out of range.	Check the ranges of the settings.	Chapter 5
223	Too much data.	Check data byte lengths.	Chapter 5
224	Illegal parameter value.	Check the ranges of the settings.	Chapter 5
225	Overflow.	Keep program messages to 1024 bytes or less in length, including <PMT>.	4-3
226	Out Of Memory.	Keep program messages to 1024 bytes or less in length, including <PMT>.	4-3
241	Hardware missing.	Check that the specified options are all installed.	—
260	Expression error.	Mathematical operations cannot be used.	—
270	Macro error.	The instrument does not support the IEEE 488.2 macro specifications.	—
272	Macro execution error.	The instrument does not support the IEEE 488.2 macro specifications.	—
273	Illegal macro label.	The instrument does not support the IEEE 488.2 macro specifications.	—
275	Macro definition too long.	The instrument does not support the IEEE 488.2 macro specifications.	—
276	Macro recursion error.	The instrument does not support the IEEE 488.2 macro specifications.	—
277	Macro redefinition not allowed.	The instrument does not support the IEEE 488.2 macro specifications.	—
278	Macro header not found.	The instrument does not support the IEEE 488.2 macro specifications.	—

Communication Query Errors (400 to 499)

Code	Message	Corrective Action	Page
410	Query INTERRUPTED.	Check the transmission and reception order.	4-3
420	Query UNTERMINATED.	Check the transmission and reception order.	4-3
430	Query DEADLOCKED.	Keep program messages to 1024 bytes or less in length, including <PMT>.	4-3
440	Query UNTERMINATED after indefinite response.	Do not write a query after *IDN? or *OPT?.	—

System Errors (Communication) (300 and 399)

Code	Message	Corrective Action	Page
300	Communication device-specific error.	Servicing is required.	—
399	Fatal error in the communication driver.	Servicing is required.	—

Communication Warnings (1)

Code	Message	Corrective Action	Page
1	*OPC/? exists in message.	Write *OPC or *OPC? at the end of program messages.	—

Other Errors (350, 390)

Code	Message	Corrective Action	Page
350	Queue overflow.	Read the error queue.	6-8
390	Communication overrun error.	Decrease the baud rate.	—

Note

Code 350 occurs when the error queue overflows. This error is only returned in response to a :STATUS:ERROR? query; it is never displayed on the screen.

Appendix 3 About the IEEE 488.2-1992 Standard

This instrument's GP-IB interface conforms to the IEEE 488.2-1992 standard. This standard specifies that the following 23 items be stated in the document. This section describes these items.

- (1) **Of the IEEE 488.1 interface functions, the subsets that are supported**
See section 3.4, "GP-IB Interface Specifications."
- (2) **The operation of the device when it is assigned an address outside the 0 to 30 range.**
The address of this instrument cannot be set to an address outside the 0 to 30 range.
- (3) **Reaction of the device when the user changes the address**
The address change is detected when the user presses UTIL and then the Remote Control soft key, and changes the address. The new address is valid until the next time it is changed.
- (4) **Device settings at power-on. The commands that can be used at power-on.**
As a basic rule, the previous settings (the settings that were in use when the instrument was turned off) are used.
There are no limitations on the commands that can be used at power-on.
- (5) **Message exchange options**
 - (a) **Input buffer size**
16384 bytes.
 - (b) **Queries that return multiple response messages**
See the example of the commands in chapter 5.
 - (c) **Queries that create response data when the command syntax is being analyzed**
All queries create response data when the command syntax is analyzed.
 - (d) **Queries that create response data during reception**
There are no queries of which the response data are created upon receiving a send request from the controller.
 - (e) **Commands that have parameters that restrict one another**
There are commands such as :CHANnel<x>:PROBe[:MODE] and :CHANnel<x>:VDIV that have parameters that place restrictions unilaterally, but there are no commands that have parameters that restrict one another.
- (6) **Items that are included in the functional or composite header elements constituting a command**
See chapters 4 and 5.
- (7) **Buffer sizes that affect block data transmission**
When block data is being transmitted, the output queue is expanded to match the size of the data that is being transmitted.
- (8) **A list of program data elements that can be used in equations and their nesting limitations**
Equations cannot be used.
- (9) **Syntax of the responses to queries**
See the example of the commands in chapter 5.
- (10) **Communication between devices that do not follow the response syntax is not supported by the instrument.**
- (11) **Size of the response data block**
1 to 999999998
- (12) **A list of supported common commands**
See section 5.38, "Common Command Group."
- (13) **Device condition after a successful calibration**
The device will be performing measurements.
- (14) **The maximum length of block data that can be used for the *DDT trigger macro definition**
Not supported.
- (15) **The maximum length of the macro label for defining macros, the maximum length of block data that can be used for the macro definition, and the process when recursion is used in macro definitions**
Macro functions are not supported.
- (16) **Reply to the *IDN? query**
See section 5.38, "Common Command Group."

(17) Size of storage area for protected user data for PUD and *PUD?

*PUD and *PUD? are not supported.

(18) The length of the *RDT and *RDT? resource names

*RDT and *RDT? are not supported.

(19) The change in the status due to *RST, *LRN?, *RCL, and *SAV

- ***RST**

See section 5.38, “Common Command Group.”

- ***RCL, *SAV, *LRN?**

These common commands are not supported.

(20) The extent of the self-test using the *TST? command

Performs the same internal memory test that is executed when the user presses UTIL and then the Self Test soft key, and executes the MEMORY test.

(21) The structure of the extended return status

See chapter 6.

(22) Whether each command is processed in an overlapped manner or sequentially

See section 4.5, “Synchronization with the Controller” and chapter 5.

(23) The description of the execution of each command

See the explanations of each command’s function in chapter 5; the Features Guide, IM DLM3054HD-01EN; and the User’s Manual, IM DLM3054HD-02EN.

Index

Symbols

	Page
<Boolean>	4-10
<Current>	4-9
<Frequency>	4-9
<NRF>	4-8, 4-10
<Time>	4-9
<Voltage>	4-9
ΔT cursor	5-95
ΔV cursor	5-92

A

	Page
abbreviated form	4-7
accumulation	5-98
ACQuire group	5-61
action-on-trigger	5-259
A Delay B trigger	5-260
alarm sound	5-112
ANALysis group	5-62
angle cursor	5-90
ASETup group	5-82
attenuation	5-85
attenuation constant	5-100
auto calibration	5-83
auto deskewing	5-68
auto measurement mode	5-152
auto off	5-254
auto scroll	5-156
auto setup	5-82, 5-162
averaging	5-61

B

	Page
backlight	5-254
beep sound	5-253
bit order	5-129
block data	4-11
brightness	5-99, 5-254
bus	5-130

C

	Page
Calc item	5-80, 5-152
CALibrate group	5-83
calibration	5-83, 5-348
CAN bus analysis	5-163
CAN bus searching	5-164
CAN Bus Trigger	5-260
CAN FD bus analysis	5-170
CAN FD bus searching	5-172
CAN FD bus trigger	5-268
CHANnel group	5-84
character data	4-10
CLEar group	5-87
clear (register)	5-348
clear trace	5-87
click sound	5-253
color	5-98
commands	4-4
common command group	5-348
common commands	4-4
COMMunicate group	5-88
communication	5-88
communication status	5-250

compression save	5-110
computation	5-133
condition register	6-7
condition register query	5-250
coupling	5-84
CURSor group	5-90
cursor measurement	5-90
cursor (XY display)	5-343
cutoff frequency	5-133
CXPI bus analysis	5-179
CXPI bus searching	5-181
CXPI bus trigger	5-277
cycle mode	5-149
cyclic statistical processing	5-151

D

	Page
data	4-8
data output	5-118
data storage (save)	5-108
data transfer rate	1-2, 2-2, 3-3
date	5-253
DCL	3-6
deadlock	4-3
decode	5-187
delay	5-140
delay between channels	5-140, 5-149
delay (trigger)	5-332
demagnetization	5-85
deskewing	5-68, 5-84, 5-130
directory	5-108
display	5-98
display format	5-99
DISPlay group	5-98
distal	5-70, 5-75, 5-150
DLM series library	ii

E

	Page
edge count	5-133
edge interval trigger	5-328
edge OR trigger	5-301
edge searching	5-156
edge trigger	5-318
envelope mode	5-61
error code query	5-250
error messages	App-2
error queue	6-8
Ethernet interface	2-2
exponential average	5-100
extended event	5-89
extended event enable register setting	5-250
extended event register	6-7

F

	Page
FFT analysis	5-100
FFT analysis result list	5-104
FFT group	5-100
file	5-108
file export (screen image)	5-126
FILE group	5-108
files, moving	5-109
filter	5-133
fine grid	5-98

Index

FlexRay bus analysis.....	5-189
FlexRay bus searching.....	5-190
FlexRay bus trigger.....	5-284
forced trigger.....	5-333
format (display).....	5-99
format (split screen).....	5-99
format (waveform data).....	5-336
format (zoom waveform display).....	5-346
front panel.....	1-1, 2-1, 3-1

G

Page

GMT.....	5-253
GONogo group.....	5-112
GP-IB interface features.....	3-2
graticule.....	5-99
grid.....	5-99
group.....	4-5
GTL.....	3-6

H

Page

hard copy.....	5-118
harmonic analysis.....	5-64
HCOPy group.....	5-118
header.....	4-7
high-pass filter.....	5-133
high resolution mode.....	5-61
histogram.....	5-62
histogram parameter.....	5-339
history.....	5-120
HISTory group.....	5-120
hold-off.....	5-333
hysteresis.....	5-131

I

Page

I2C bus analysis.....	5-196
I2C bus searching.....	5-198
I2C bus trigger.....	5-291
IFC.....	3-6
IIR filter.....	5-133
IMAGe group.....	5-126
initialization.....	5-128, 5-349
INITialize group.....	5-128
input coupling.....	5-84
input filter.....	5-84
integration (MATH).....	5-134
integration (XY display).....	5-344
internal memory.....	5-154
interpolation method.....	5-99
interval trigger.....	5-328
inversion.....	5-84
inverted display.....	5-84

J

Page

Joule integral.....	5-66
judgment.....	5-112

L

Page

label.....	5-84, 5-130, 5-134
LCD.....	5-254
LIN bus analysis.....	5-204
LIN bus searching.....	5-205
LIN bus trigger.....	5-296
linear scaling.....	5-85
LLO.....	3-6
load (recall).....	5-109
local lockout.....	5-88

lockout.....	5-88
LOGic group.....	5-129
logic input.....	5-129
low-pass filter.....	5-134

M

Page

magnification.....	5-346
mail notification.....	5-112
marker cursor.....	5-93
marker cursor (FFT analysis).....	5-102
MATH group.....	5-133
MEASure group.....	5-139
measurement, text size.....	5-255
menu language.....	5-254
mesial.....	5-70, 5-75, 5-150
message.....	4-1
message language.....	5-254
message (top of the error queue) query.....	5-250
minimum record number.....	5-337
model.....	5-348
moving average.....	5-134
multiple input combination searching.....	5-157
multiple input combination trigger.....	5-301
multi target.....	5-118

N

Page

normal mode.....	5-61

O

Page

offset.....	5-85
offset voltage, application.....	5-255
operators.....	5-135
option query.....	5-349
output queue.....	6-8
overlap commands.....	4-12

P

Page

parameters.....	5-139, 5-147
path (directory).....	5-109
pattern searching.....	5-157
pattern trigger.....	5-301
physical value conversion.....	5-336
position (vertical).....	5-85, 5-132, 5-155
power measurement.....	5-75
power supply analysis.....	5-62
power supply analysis input.....	5-68
printer.....	5-119
probe attenuation.....	5-85
program messages.....	4-1
proximal.....	5-70, 5-75, 5-150
PSI5 analysis.....	5-209
PSI5 searching.....	5-212
PSI5 trigger.....	5-302
pulse width searching.....	5-159
pulse width trigger.....	5-329

Q

Page

query.....	4-1, 4-6

R

Page

rear panel.....	1-1, 2-1
RECall group.....	5-154
recall (load).....	5-109
record length.....	5-61
reference.....	5-155
REference group.....	5-155

register, clearing	5-348
remote	5-88
remote and local modes, switching	1-3, 2-3, 3-3
REN	3-6
rename	5-109
response messages	4-2
responses	4-7
rise/fall time trigger	5-329
rotary count	5-135
run trigger	5-330

S	Page
safe operating area analysis	5-69
sample rate	5-258, 5-337
sampling mode	5-61
save	5-108
save (store)	5-252
scale value display	5-99
scaling coefficient	5-85
screen data, output	5-118
screen image	5-126
SDC	3-6
SEARCh group	5-156
search (history feature)	5-120
search type	5-161
search type (serial bus)	5-242
self-test	5-350
SENT analysis	5-217
SENT searching	5-221
SENT trigger	5-307
SERialbus group	5-162
serial bus signals, analyzing and searching	5-162
serial polling execution	5-250
service request enable register value setting	5-349
settings, initialization	5-349
side panel	3-1
sign inclusion	5-337
single start	5-248
skip	5-160
SNAP group	5-247
snapshot	5-247
SNTP	5-253
SPD	3-6
SPE	3-6
SPI bus analysis	5-237
SPI bus searching	5-240
SPI bus trigger	5-322
split screen format	5-99
SSTart group	5-248
standard event register	6-5
STARt group	5-249
start (waveform acquisition)	5-249, 5-251
state display	5-98, 5-131
statistical processing	5-141, 5-147
status	5-250
status byte	6-3
status byte register query	5-349
STATus group	5-250
status reports	6-1
STOP group	5-251
storage	5-108
storage device	5-108
STORe group	5-252
string data	4-10
switching loss	5-70
system	5-253
SYSTem group	5-253

Index

T	Page
TCP/IP Settings	2-6
threshold level	5-129
time	5-253
time base	5-258
TIMEbase group	5-258
Time/div value	5-258
timeout (ANALysis)	5-81
timeout (GO/NO-GO)	5-113
timeout (measure)	5-153
timeout period searching	5-160
timeout period, setting	2-3
timeout period trigger	5-330
timeout trigger	5-330
TMCTL	ii
top-of-error-queue query	5-250
total loss	5-70
transition filter	5-250, 6-7
trigger	5-259
A->B(N)	5-259
A Delay B	5-260
CAN bus	5-260
CAN FD bus	5-268
CXPI bus	5-277
edge	5-318
edge interval	5-328
edge OR	5-301
FlexRay bus	5-284
forced	5-333
I2C bus	5-291
interval	5-328
LIN bus	5-296
multiple input combination	5-301
pattern	5-301
PSI5	5-302
pulse width	5-329
rise/fall time	5-329
Rise/Fall Time	5-329
runt	5-330
Runt	5-330
SENT	5-307
SPI bus	5-322
Timeout	5-330
timeout period	5-330
TV	5-330
UART	5-325
user-defined bus	5-320
user-defined TV	5-331
window	5-328
window OR	5-328
trigger combination	5-332
trigger conditions	5-260
TRIGger group	5-259
trigger output	5-255
trigger position	5-333
trigger slope	5-328, 5-329
trigger type setting	5-325
TV trigger	5-330

U	Page
UART analysis	5-242
UART searching	5-244
UART trigger	5-325
upper-level query	4-6
USB cable	1-4
USB hub	1-4

Index

USB interface	1-2
USB keyboard type.....	5-257
user-defined bus analysis.....	5-233
user-defined bus searching	5-236
user-defined bus trigger.....	5-320
user-defined TV trigger.....	5-331

V **Page**

vertical axis (vertical axis) sensitivity	5-86
vertical axis (voltage axis)	5-84
vertical position.....	5-85, 5-132, 5-155

W **Page**

waveform acquisition	5-61
waveform data format.....	5-336
waveform data setting	5-335
WAVeform group.....	5-335
waveform parameter measurement.....	5-339
waveform parameters.....	5-139, 5-147
waveform searching	5-156
window function	5-107
window OR trigger.....	5-328
window trigger	5-328
WPARameter group.....	5-339

X **Page**

XY display.....	5-343
XY group.....	5-343

Z **Page**

zero calibration	5-85
zoom.....	5-346
zoom box	5-346
ZOOM group.....	5-346
zoom position	5-347
zoom waveform display format.....	5-346