WT1801R, WT1802R, WT1803R, WT1804R, WT1805R, WT1806R

Precision Power Analyzer Communication Interface

USER'S MANUAL

YOKOGAWA
Yokogawa Test & Measurement Corporation

IM WT1801R-17EN
1st Edition

Thank you for purchasing the WT1801R, WT1802R, WT1803R, WT1804R, WT1805R, or WT1806R Precision Power Analyzer.
This Communication Interface User's Manual explains the following interface features and commands.

- Ethernet interface
- USB interface
- GP-IB interface

To ensure correct use, please read this manual thoroughly before operation.
Keep this manual in a safe place for quick reference in the event 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, MS-DOS, Windows, Windows 10, and Windows 11 are either 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.
- Modbus is a registered trademark of Schneider Electric USA, Inc.
- 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.

1st Edition: October 2024 (YMI)

About the USB interface and Ethernet interface

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

- Communication library (TMCTL)

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

- Communication library (TMCTL)
- USB device driver for connecting this instrument to the PC

To download the libraries and drivers listed above, go to the following website, and then browse to the download page.
https://tmi.yokogawa.com/

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
WT1801R, WT1802R, WT1803R, WT1804R, WT1805R, WT1806R Precision Power Analyzer Getting Started Guide	IM WT1801R-03EN	Explains the handling precautions and basic operations of this instrument.
WT1801R, WT1802R, WT1803R, WT1804R, WT1805R, WT1806R Precision Power Analyzer Request to Download Manuals	IM WT1801R-73Z2	Describes the manuals provided on the website.
WT1801R, WT1802R, WT1803R, WT1804R, WT1805R, WT1806R Precision Power Analyzer	IM WT1801R-92Z1	Document for China
Safety Instruction Manual	IM 00C01C01-01Z1	Safety manual (European languages)

Manuals provided on the website

Download the following manuals from the YOKOGAWA website.

Manual Title	Manual No.	Description
WT1801R, WT1802R, WT1803R, WT1804R, WT1805R, WT1806R Precision Power Analyzer Features Guide	IM WT1801R-01EN	Explains all the instrument's features other than the communication interface features.
WT1801R, WT1802R, WT1803R, WT1804R, WT1805R, WT1806R Precision Power Analyzer User's Manual	IM WT1801R-02EN	Explains how to operate this instrument.
WT1801R, WT1802R, WT1803R, WT1804R, WT1805R, WT1806R Precision Power Analyzer Communication Interface User's Manual	IM WT1801R-17EN	This document. It 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 WT1801R-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.

How to Use This Manual

Structure of the manual

This manual contains eight chapters and an appendix.

Chapter 1 Ethernet Interface

Describes the features and specifications of the Ethernet interface.

Chapter 2 USB Interface

Describes the features and specifications of the USB interface.

Chapter 3 GP-IB Interface

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.

Chapter 7 Modbus/TCP Communication

Provides an overview of Modbus/TCP communication and describes registers and the like.

Chapter 8 Commands Compatible with Legacy Instruments

Describes the commands compatible with legacy models (WT1600, WT1800, and WT1800E).

Appendix

Describes error messages and provides other information.

Index

Conventions Used in This Manual

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.

French

AVERTISSEMENT Attire l'attention sur des gestes ou des conditions susceptibles de provoquer des blessures graves (voire mortelles), et sur les précautions de sécurité pouvant prévenir de tels accidents.

ATTENTION Attire l'attention sur des gestes ou des conditions susceptibles de provoquer des blessures légères ou d'endommager l'instrument ou les données de l'utilisateur, et sur les précautions de sécurité susceptibles de prévenir de tels accidents.

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

Character notations

Panel key names and soft key names in bold characters

Indicate panel keys that are used in the procedure and soft keys and menu items that appear on the screen.

SHIFT+panel key

When SHIFT+panel key appears in a procedural explanation, it means to press the shift key so that it illuminates, and then to press the indicated panel key. A setup menu for the item written in purple below the key that you pressed appears on the screen.

Unit

k Denotes 1000. Example: 100 kHz

K Denotes 1024. Example: 720 KB (file size)

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, see pages 4-8 to 4-10.

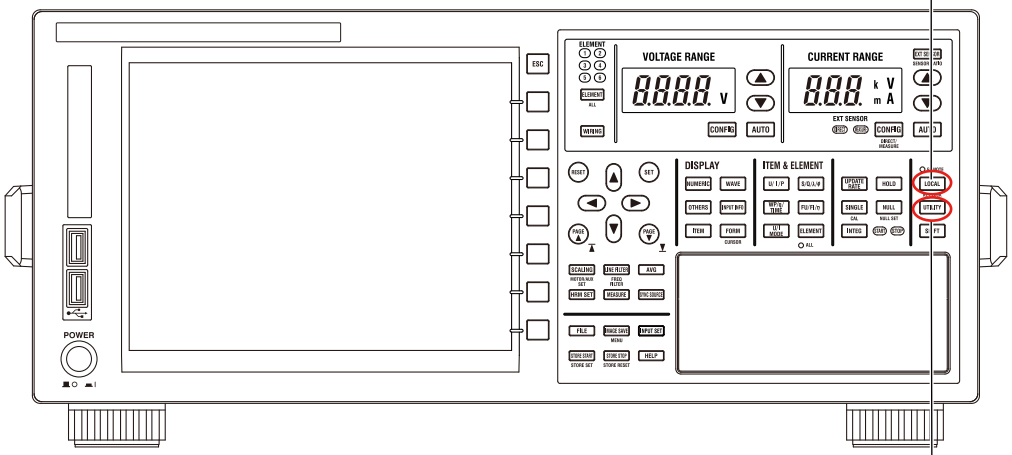
Symbol	Description	Example	Example of Input
< >	A defined value	`ELEMent<x>` <x> = 1 to 6	`ELEMENT2`
{ } |	Select an option in { } Exclusive OR	`SQFormula {TYPE1|TYPE2|TYPE3}`	`SQFORMULA TYPE1`
[ ]	Can be omitted	`NUMeric[:NORMal]:VALue?`	`NUMERIC:VALUE?`

Contents

1.1 Component Names and Functions

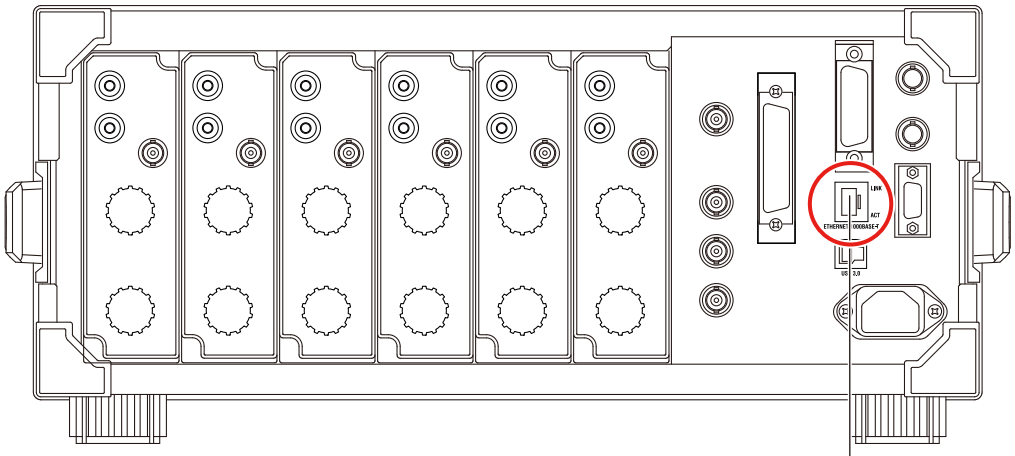
Front panel

LOCAL key
Press this key 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.
This key is disabled when local lockout (see page 1-2) has been activated by a controller.



UTILITY key (page 1-5)
Press this key to set the network connection timeout setting.

Rear panel



Ethernet port
This port is for connecting the instrument to a controller (such as a PC) using an Ethernet cable. For details on how to connect the instrument to a controller, see page 1-4.

1.2 Ethernet Interface Features and Specifications

Ethernet interface features

Reception feature

You can use the reception feature to specify the same settings that you can specify by using the front panel keys.
This 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
Communication protocol	TCP/IP (VXI-11)
Connector type	RJ-45

Switching between remote and local modes

Switching from local to remote mode

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

- The REMOTE indicator illuminates.
- All keys except the **LOCAL** key are disabled.
- Local mode settings are retained even when this instrument switches to remote mode.

Switching from remote to local mode

When this instrument is in remote mode and you press **LOCAL**, this instrument switches to local mode. However, this does not work if this 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 turns off.
- 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).

Setting the timeout value

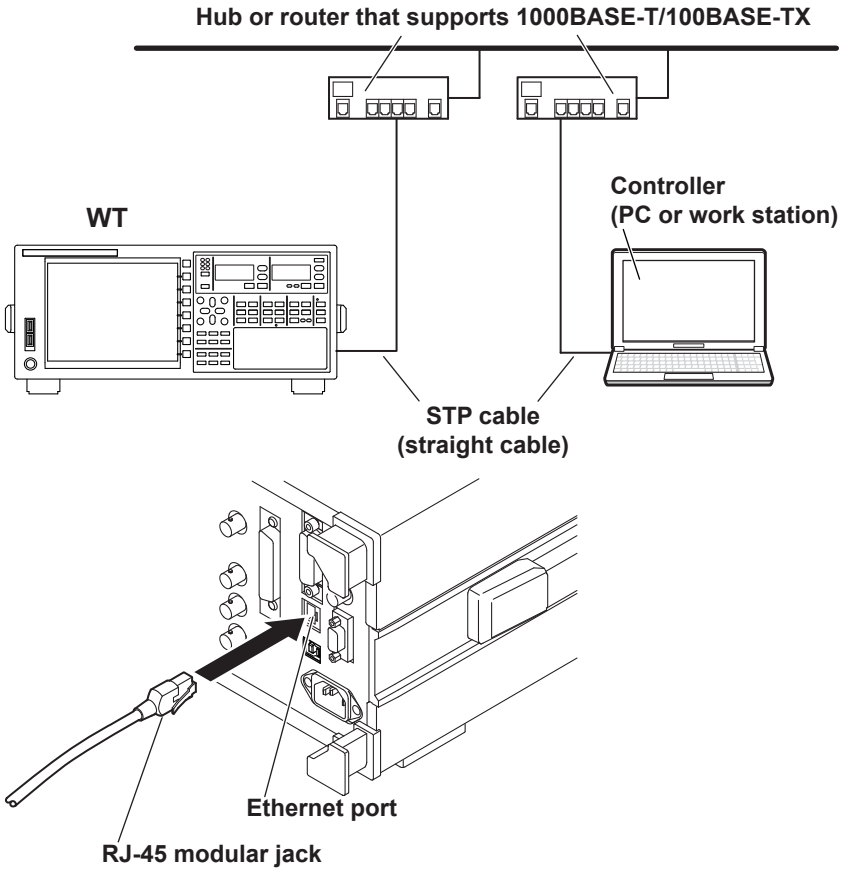
If the instrument is not accessed for a specified amount of time (set as a timeout period), the instrument closes the connection to the network. The timeout value can be set to Infinite or in the range of 1 to 3600 s. The default value is Infinite.

For instructions on how to set the timeout value, see section 1.4, “Configuring the Instrument’s Ethernet Settings.”

1.3 Ethernet Interface Connection

Connection procedure

Connect an STP (shielded twisted-pair) 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 20.1, "Connecting the Instrument to a Network" in the *User's Manual*, IM WT1801R-02EN.

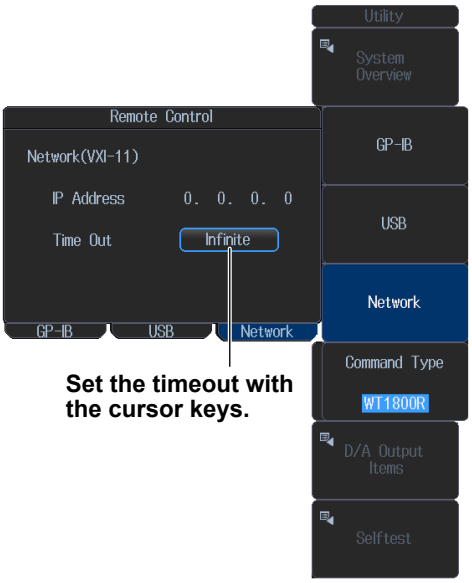
1.4 Configuring the Instrument's Ethernet Settings

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

- Network connection timeout setting

UTILITY Remote Control menu

Press **UTILITY**, the **Remote Control** soft key, and then the **Network** soft key to display the following screen.



Note

Only use one communication interface: GP-IB, USB, or Network. If you send commands simultaneously from more than one communication interface, the instrument will not execute the commands properly.

Configuring the TCP/IP settings

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

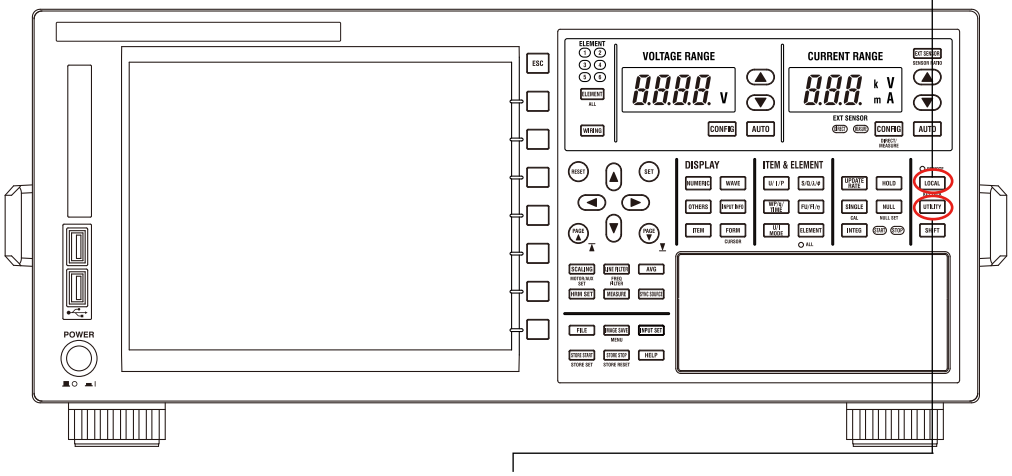
- IP address
- Subnet mask
- Default gateway

For instructions on how to specify these settings, see section 20.2, "Configuring the TCP/IP Settings" in the *User's Manual*, IM WT1801R-02EN.

2.1 Component Names and Functions

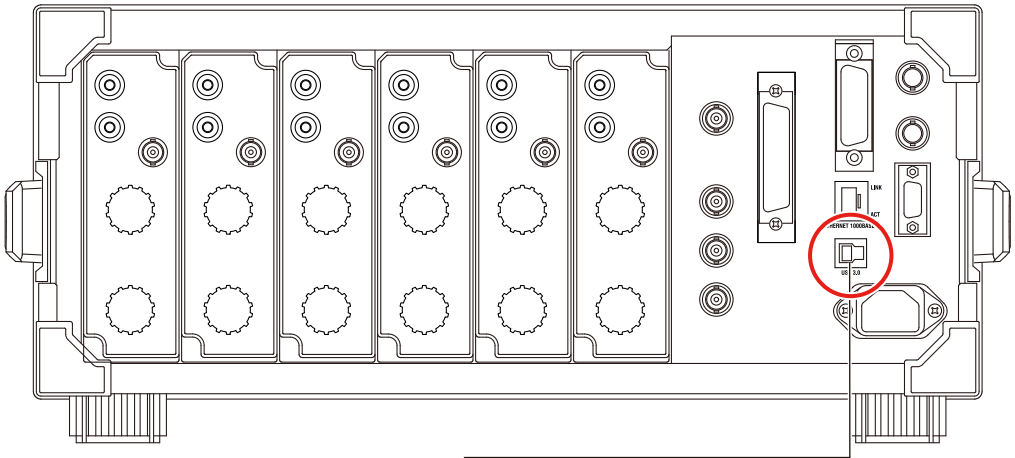
Front panel

LOCAL key
Press this key 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.
This key is disabled when local lockout (see page 2-2) has been activated by a controller.



UTILITY key (page 2-4)
Press this key to view the serial number that is to be used in USB TMC communication.

Rear panel



USB port
This port is for connecting the instrument to a controller (such as a PC) using a USB cable.
For details on how to connect the instrument to a controller, see page 2-3.

2.2 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.
This 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 compliant
Connector	Type B connector (receptacle)
Number of ports	1
Power supply	Self powered
System requirements	A PC with a standard USB port, running Windows 10 or Windows 11. A separate device driver is required to enable the connection with the PC.

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 it receives a `:COMMunicate:REMote ON` command from the PC.

- The REMOTE indicator illuminates.
- All keys except the **LOCAL** 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 **LOCAL**, 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 turns off.
- All keys are enabled.
- Remote mode settings are retained even when the instrument switches to local mode.

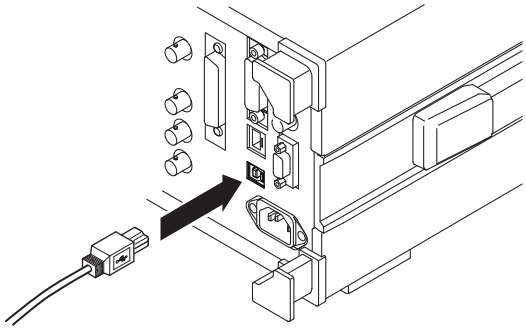
Note

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

2.3 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 or remove USB cables from the time when this instrument is turned on until operation becomes available (approximately 20 to 30 seconds). Doing so may damage the instrument.



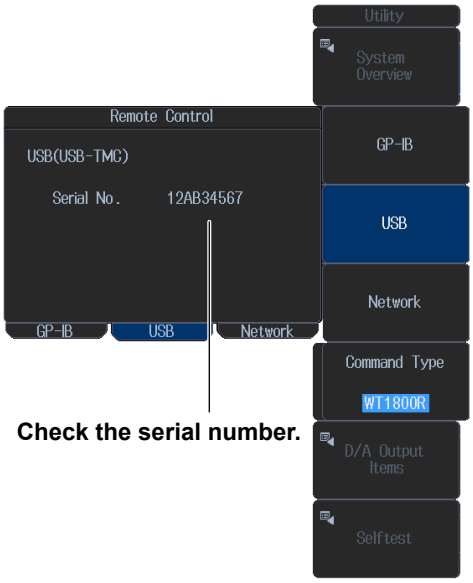
2.4 Configuring the Instrument's USB Settings

This section explains the following setting for controlling the instrument remotely through a USB interface:

- Checking the serial number that is used in USB TMC communications

UTILITY Remote Control menu

Press **UTILITY**, the **Remote Control** soft key, and then the **USB** soft key to display the following screen.



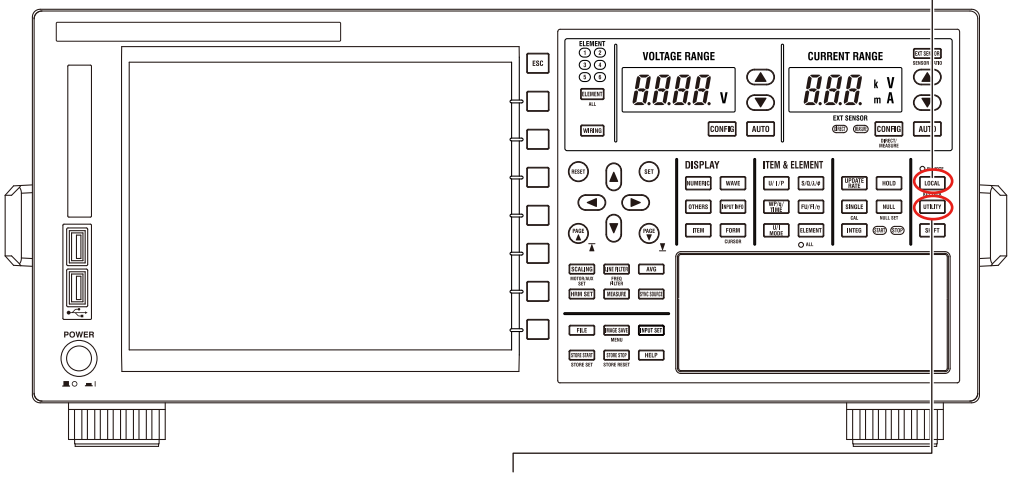
Note

- Only use one communication interface: GP-IB, USB, or Network. If you send commands simultaneously from more than one communication interface, the instrument will not execute the commands properly.
- Install the YOKOGAWA USB TMC (Test and Measurement Class) driver on your PC. For information about how to obtain the YOKOGAWA USB TMC driver, contact your nearest YOKOGAWA dealer. You can also access the YOKOGAWA USB driver download web page and download the driver.
  https://tmi.yokogawa.com/
- Do not use USB TMC drivers (or software) supplied by other companies.

3.1 Component Names and Functions

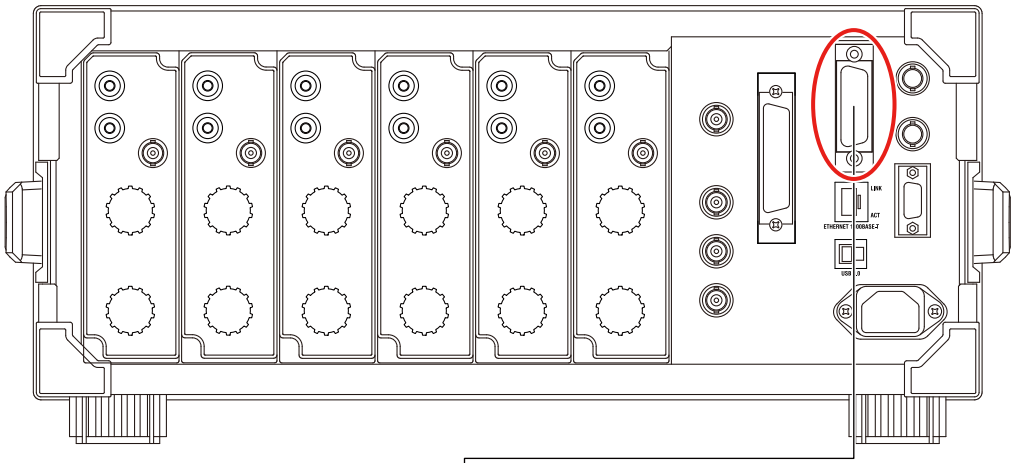
Front panel

LOCAL key
Press this key 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.
This key is disabled when local lockout (see page 3-7) has been activated by a controller.



UTILITY key (page 3-6)
Press this key to set the GP-IB address.

Rear panel



GP-IB port
This port is for connecting this instrument to a controller (such as a PC) using a GP-IB cable. For details on how to connect this instrument to a controller, see page 3-4.

3.2 GP-IB Interface Features and Specifications

GP-IB interface features

Listener capabilities

- Allows you to specify the same settings that you can specify by using the front panel keys. This excludes turning the power on and off and changing the communication settings.
- Receives 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

Supported devices	National Instruments Corporation • PCI-GPIB and PCI-GPIB+ • PCIe-GPIB and PCIe-GPIB+ • PCMCIA-GPIB and PCMCIA-GPIB+ • GPIB-USB-HS • GPIB-USB-HS+ Driver NI-488.2M Version 1.60 or later
Electrical and mechanical	Complies with IEEE St'd 488-1978
Functional specifications	See the table below.
Protocol	Complies with IEEE St'd 488.2-1992
Code	ISO (ASCII) code
Mode	Addressable mode
Address setting	Press **UTILITY** and then the **Remote Control** soft key. Then, set the network interface (Device) to GP-IB and the address to a number from 0 to 30.
Clearing remote mode	Clear remote mode by pressing **LOCAL** 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, and untalk on MLA (My Listen Address). 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 poll capability
Device clear	DC1	Full device clear capability
Device trigger	DT1	Device trigger capability
Controller	C0	No controller capability
Electric characteristics	E1	Open collector

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 it receives a REN (Remote Enable) message from the PC.

- The REMOTE indicator illuminates.
- All keys except the **LOCAL** 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 **LOCAL**, the instrument switches to local mode. This key is disabled when local lockout (see page 3-7) has been activated by a controller.

- The REMOTE indicator turns off.
- 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 this instrument before you connect or remove communication cables. Otherwise, erroneous operation may result, or the internal circuitry may break.

French

ATTENTION

Veiller à mettre le PC et l'instrument hors tension avant de brancher ou de débrancher les câbles de communication, pour éviter de provoquer des dysfonctionnements ou des courts-circuits internes.

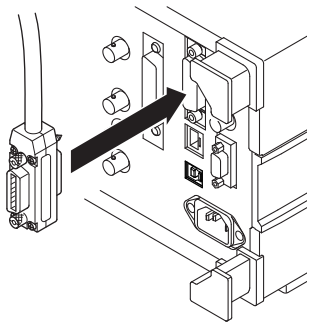
GP-IB cable

The instrument is equipped with an IEEE St'd 488-1978 24-pin GP-IB connector. Use GP-IB cables that conforms to this standard.

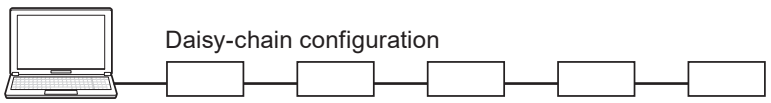
Connection procedure

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

This instrument (rear panel)

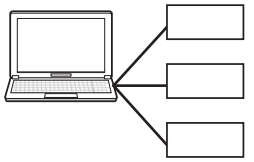


Controller



Daisy-chain configuration

Star configuration



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 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 following setting, which needs to be configured when controlling the instrument remotely through a GP-IB interface.

- Setting the GP-IB address

UTILITY Remote Control menuS

Press **UTILITY**, the **Remote Control** soft key, and then the **GP-IB** soft key to display the following screen.

Note

- Only use one communication interface: GP-IB, USB, or Network. If you send commands simultaneously from more than one communication interface, 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 page 6-6).
- Discards `*OPC` and `*OPC?` commands that are being executed.
- Immediately aborts `*WAI` and `:COMMunicate:WAIT` commands.

GET (Group Execute Trigger)

The same operation as the `*TRG` command.

PPC (Parallel Poll Configure) and TCT (Take Control) are not supported.

Responses to multi-line messages (universal commands)

LLO (Local Lockout)

Prohibits switching to local mode by disabling the LOCAL key on the front panel.

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 grouped 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. They 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

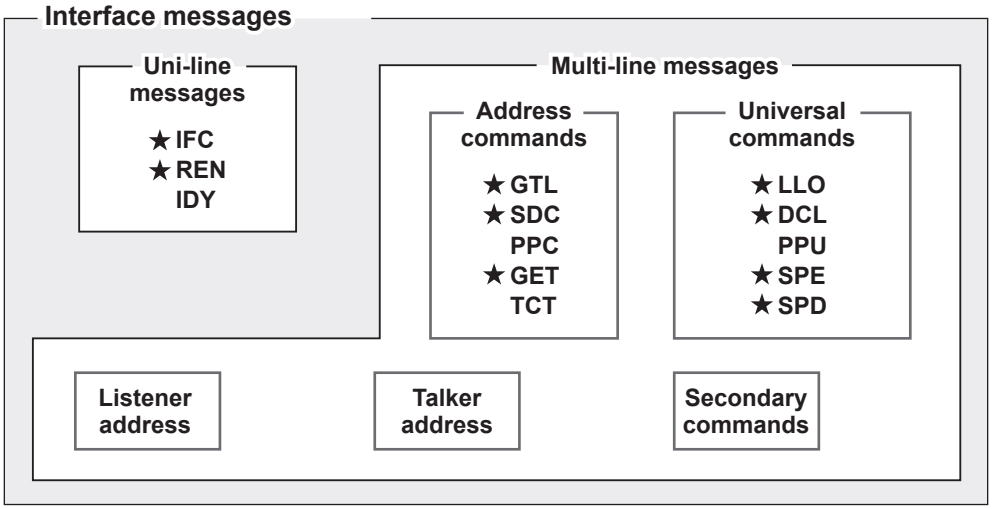
- TCT (Take Control)

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)

There are other interface messages: listener-address, talk-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

Information is exchanged between a controller and this instrument in units of messages. 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 (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 syntax 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 `:INPut:CFACtor 3;INDependent OFF<PMT>`

Unit Unit

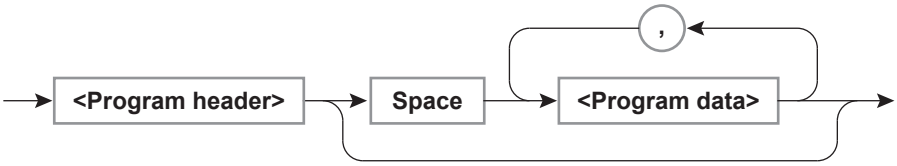
<PMT>

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

NL (new line)	Same as LF (line feed). ASCII code “0AH”
^END	The END message as defined by IEEE 488.1. The data byte that is sent with the END message is the last data byte of the program message.
NL^END	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.



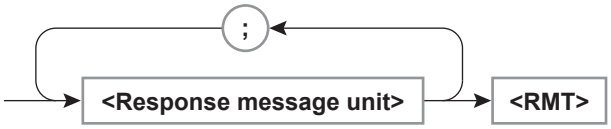
<Program header>	The program header indicates the command type. For details, see page 4-4.
<Program data>	Attach program data if there are conditions that are required to execute a command. Separate the program data from the header with a space (ASCII code 20H). If there are multiple sets of program data, separate each set with a comma. For details, see page 4-8.

Example `:INPut:CFACtor 3<PMT>`

Header **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 `:INPUT:CFACTOR 3;INDEPENDENT 0<RMT>`

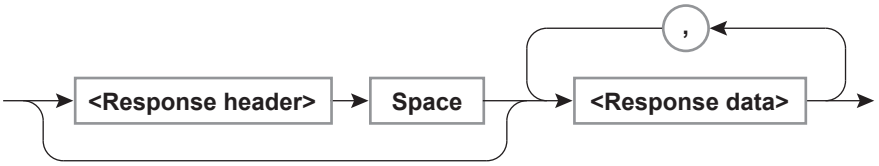
Unit **Unit**

<RMT>

RMT is a response message terminator. It is `NL^END`.

Response message unit syntax

The response message unit syntax is shown below.



<Response header>	A response header sometimes precedes the response data. Separate the data from the header with a space. For details, see page 4-7.
<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 page 4-7.

Example `100.00E-03<RMT>` `:DISPLAY:MODE WAVE<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 nth response unit may not necessarily correspond to the nth 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, this 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 below 1024 bytes. Program messages that do not contain queries never cause deadlocks.

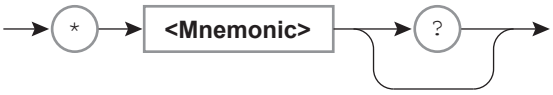
4.2 Commands

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

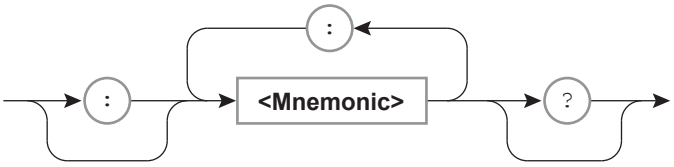
Commands that are defined in IEEE 488.2-1992 are called common commands. The common command header syntax is shown below. Be sure to include an asterisk (*) at the beginning of a common command.



Common command example: **`*CLS`**

Compound header

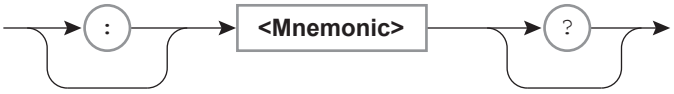
Other commands that are specific to the instrument are classified and arranged in a hierarchy according to their functions. The compound header syntax is shown below. Be sure to use a colon to specify a lower hierarchical level.



Compound header example: **`:DISPlay: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: **`:HOLD`**

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 A portion of the commands from the integration command group

- `:INTEGrate?`
- `:INTEGrate:MODE`
- `:INTEGrate:TIMer`
- `:INTEGrate:RTIMe?`
- `:INTEGrate:RTIMe:STARt`
- `:INTEGrate:RTIMe:END`
- `:INTEGrate:ACAL`
- `:INTEGrate:STARt`
- `:INTEGrate:STOP`
- `:INTEGrate:RESet`

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 `:INTEGrate:MODE NORMal;ACAL ON<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 `:INTEGrate:MODE NORMal;:DISPlay:MODE NUMeric<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 `:INTEGrate:MODE NORMal;:HOLD ON<PMT>`

Concatenating common commands

Common commands that are defined in IEEE 488.2-1992 are independent of hierarchy. A colon is not needed before a common command.

Example `:INTEGrate:MODE NORMal;*CLS;ACAL ON<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 `:INTEGrate:MODE NORMal<PMT>:INTEGrate:ACAL ON<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 **`:INTEGrate?<PMT> -> :INTEGRATE:MODE NORMAL;TIMER 0,0,0;ACAL 0<RMT>`**

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 **`ELEM`** for **`ELEMent<x>`**, **`ELEMent1`** is specified.

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

  Example **`[:INPut]SCALing[:STATe][:ALL] ON`** can be written as **`SCAL ON`**.

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

  Example **`SCALing?`** and **`SCALing:STATe?`** 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 **:DISPlay:MODE?<PMT> -> :DISPLAY:MODE WAVE<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 **[:INPut]:POVer?<PMT> -> 0<RMT>**

Receiving 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:VERBose` 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 classified as follows:

Group	Meaning	Page
<Decimal>	A value expressed in decimal notation Example: VT ratio setting `->[:INPut]:SCALing:VT:ELEMent1 100`	4-8
<Voltage>, <Current>, <Time>, <Frequency>	A physical value Example: Voltage range setting `->[:INPut]:VOLTage:RANGE:ELEMent1 100V`	4-9
<Register>	A register value expressed as binary, octal, decimal, or hexadecimal Example: Extended event register value `->:STATUS:EESE #HFE`	4-9
<Character data>	Predefined character string (mnemonic). Select from the available strings in braces. Example: Trigger mode selection `->:DISPlay:WAVE:TRIGger:MODE {AUTO|NORMal|OFF}`	4-10
<Boolean>	Indicates on and off. Specify ON, OFF, or a value. Example: Turning data hold on `->:HOLD ON`	4-10
<String data>	User-defined string Example: User-defined function `->:MEASure:FUNCtion1:EXPRession "URMS(E1)"`	4-10
<Filename>	Indicates a file name. Example: Save file name `->:FILE:SAVE:SETup[:EXECute] "CASE1"`	–
<Block data>	Data that contains 8-bit values Example: Response to acquired waveform data `-> #40012ABCDEFGHIJKL`	4-10

<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	Meaning	Example		
<NR1>	Integer	`125`	`-1`	`+1000`
<NR2>	Fixed-point number	`125.0`	`-.90`	`+001.`
<NR3>	Floating-point number	`125.0E+0`	`-9E-1`	`+.1E4`
<NRf>	Any form from <NR1> to <NR3>			

- The instrument can receive decimal values that are sent from the controller in any of the froms <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>, <Current>, <Time>, <Frequency>

<Voltage>, <Current>, <Time>, and <Frequency> 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><Multiplier>	5M
<NRf>	5E-3

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

<Multiplier>

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

<Unit>

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

- <Multiplier> and <Unit> are not case sensitive.
- “U” is used to indicate micro (“μ”).
- “MA” is used for Mega to distinguish it from Milli. However, “MA” is interpreted as milliampere for current. In addition, megahertz is expressed as “MHZ.” Therefore, “M (Milli)” cannot be used for frequencies.
- If both <Multiplier> and <Unit> are omitted, the basic unit (V, A, S, or HZ) is used.
- Response messages are always expressed in the <NR3> form. Additionally, they are returned in the basic form, without a multiplier or unit attached.

<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
{AUTO|NORMal}	AUTO

- 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. The following types of expressions are possible.

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

- When expressed 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' "IEEE488.2-1992"

- 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. No error is detected if a quotation mark is omitted.

<Block data>

<Block data> is any 8-bit data. It is only used in response messages on this 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. With overlap commands, the execution of the next command may start before the execution of the previous command is completed.
If you specify a voltage range and send the next program message to query for the result, the instrument always returns the most recent setting (100 V in this case).

```
:INPut:VOLTage:RANGe:ELEMent1 100V;ELEMent1?<PMT>
```

This is because the next command is forced to wait until the processing of `:INPut:VOLTage:RANGe:ELEMent1` is completed. This type of command is called a sequential command.
On the other hand, for example, if you execute a file load, query for the voltage range value, and send the next program message, `:INPut:VOLTage:RANGe:ELEMent1?` is executed before the loading of the file is completed, and the returned voltage range value is the value before the file is loaded.

```
:FILE:LOAD:SETup "FILE1";:INPut:VOLTage:RANGe:ELEMent1?
```

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`.
You can prevent overlapping by using the following methods.

Synchronizing to overlap commands

Using the *WAI command

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

Example
```
:COMMunicate:OPSE #H0040;:FILE:LOAD:SETup
"FILE1";*WAI;:INPut:VOLTage:RANGe:ELEMent1?<PMT>
```

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 `:INPut:VOLTage:RANGe:ELEMent1?`,
so `:INPut:VOLTage:RANGe:ELEMent1?` is not executed until the file loading is completed.

Using the :COMMunicate:OVERlap command

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

Example **:COMMunicate:OVERlap #HFFBF;:FILE:LOAD:SETup "FILE1";:INPut:VOLTage:RANGe:ELEMent1?<PMT>**

:COMMunicate:OVERlap #HFFBF enables overlapping for commands other than media access. Because overlapping of file loading is disabled, :FILE:LOAD:SETup operates in the same way as a sequential command. Thus, :INPut:VOLTage:RANGe:ELEMent1? is not executed until 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 page 6-5), to 1 when the overlapping is completed.

Example **:COMMunicate:OPSE #H0040;*ESE 1;*ESR?;*SRE 32**
;:FILE:LOAD:SETup "FILE1";*OPC<PMT>
(Read the response to*ESR?)
(Wait for a service request)
:INPut:VOLTage:RANGe:ELEMent1?<PMT>

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.
In the example above, :INPut:VOLTage:RANGe:ELEMent1? is not executed until a service request is generated.

Using the *OPC? query

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

Example **:COMMunicate:OPSE #H0040;:FILE:LOAD:SETup "FILE1";*OPC?<PMT>**
(Read the response to *OPC?)
:INPut:VOLTage:RANGe:ELEMent?<PMT>

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.

Note

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

Achieving synchronization without using overlap commands

Even when using sequential commands, there are times when it is necessary to achieve synchronization to properly query the measured data. For example, if you want to query the most recent numeric data each time that the measured data is updated, you can attempt to do this by sending the :NUMeric[:NORMal]:VALue? command with some arbitrary timing. However, because the instrument returns the current measured data regardless of whether the measured data has been updated since the previous query, this method may return data that is the same as the previous data. If this happens, you must use the following method to synchronize with the end of measured data updating.

Using the :STATus:CONDition? query

:STATus:CONDition? is used to query the contents of the condition register (see page 6-7). You can determine whether the measured data is being updated by reading bit 0 of the condition register. If bit 0 of the condition register is 1, the measured data is being updated. If it is 0, the measured data can be queried.

Using the extended event register

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

Example **:STATus:FILTer1 FALL;:STATus:EESE 1;EESR?;*SRE 8<PMT>**
(Read the response to :STATus:EESR?)
Loop
(Wait for a service request)
:NUMeric[:NORMal]:VALue?<PMT>
(Read the response to :NUMeric[]NORMal]:VALue?)
:STATus:EESR?<PMT>
(Read the response to :STATus:EESR?)
(Return to Loop)

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:EESE 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.
The :NUMeric[:NORMal]:VALue? command 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.

Example **:STATus:FILTer1 FALL;:STATus:EESR?<PMT>**
(Read the response to :STATus:EESR?)
Loop
:COMMunicate:WAIT 1<PMT>
:NUMeric[:NORMal]:VALue?<PMT>
(Read the response to :NUMeric[]NORMal]:VALue?)
:STATus:EESR?<PMT>
(Read the response to :STATus:EESR?)
(Return to Loop)

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.
:NUMeric[:NORMal]:VALue? is not executed until bit 0 in the extended event register is set to 1.

5.1 List of Commands

Command	Explanation	Page
AOUTput Group		
:AOUTput?	Queries all D/A output settings.	5-22
:AOUTput:NORMal?	Queries all D/A output settings.	5-22
:AOUTput[:NORMal]:CHANnel<x>	Sets or queries a D/A output item (function, element, or harmonic order).	5-22
:AOUTput[:NORMal]:IRTime	Sets or queries the integration time that is used in the D/A output of the integrated value.	5-22
:AOUTput[:NORMal]:MODE<x>	Sets or queries the rated-value setup mode for D/A output items.	5-22
:AOUTput[:NORMal]:RATE<x>	Sets or queries the rated maximum or minimum value for D/A output items.	5-22
AUX Group		
:AUX<x>?	Queries all auxiliary input settings.	5-23
:AUX<x>:AUTO	Sets or queries the voltage auto range on/off status of the specified auxiliary input.	5-23
:AUX<x>:FILTer?	Queries all input filter settings for the auxiliary inputs.	5-23
:AUX<x>:FILTer[:LINE]	Sets or queries the line filter for the auxiliary inputs.	5-23
:AUX<x>:LSCale?	Queries all auxiliary input linear scaling settings.	5-23
:AUX<x>:LSCale:AVALue	Sets or queries the slope (A) of the linear scale of the auxiliary input feature.	5-23
:AUX<x>:LSCale:BVALue	Sets or queries the offset (B) of the linear scale of the auxiliary input feature.	5-23
:AUX<x>:LSCale:CALCulate?	Queries all parameter calculation settings for the linear scale of the auxiliary input feature.	5-23
:AUX<x>:LSCale:CALCulate:{P1X|P1Y|P2X|P2Y}	Sets or queries the data (Point1X, Point1Y, Point2X, or Point2Y) for parameter calculations of the linear scale of the auxiliary input feature.	5-23
:AUX<x>:LSCale:CALCulate:EXECute	Calculates parameters for the linear scale of the auxiliary input feature.	5-24
:AUX<x>:NAME	Sets or queries the auxiliary input name.	5-24
:AUX<x>:RANGe	Sets or queries the auxiliary input voltage range.	5-24
:AUX<x>:SCALing	Sets or queries the auxiliary input scaling factor.	5-24
:AUX<x>:UNIT	Sets or queries the unit to assign to the auxiliary input.	5-24
COMMunicate Group		
:COMMunicate?	Queries all communication settings.	5-25
:COMMunicate:HEADer	Sets or queries whether a header is added to the response to a query. (Example with header: ":DISPLAY:MODE NUMERIC." Example without header: "NUMERIC.")	5-25
:COMMunicate:LOCKout	Sets or clears local lockout.	5-25
:COMMunicate:OPSE	Sets or queries the overlap command that is used by the *OPC, *OPC?, and *WAI commands.	5-25
:COMMunicate:OPSR?	Queries the operation pending status register.	5-25
:COMMunicate:OVERlap	Sets or queries the commands that operate as overlap commands.	5-25

Command	Explanation	Page
:COMMunicate:REMote	Sets this instrument to remote or local mode. On is remote mode.	5-26
:COMMunicate:VERBose	Sets or queries whether the response to a query is returned fully spelled out (example: ":INPUT:VOLTAGE:RANGE:ELEMENT1 1.000E+03") or in its abbreviated form (example: "VOLT:RANG:ELEM 1.000E+03").	5-26
:COMMunicate:WAIT	Waits for a specified extended event to occur.	5-26
:COMMunicate:WAIT?	Creates the response that is returned when a specified extended event occurs.	5-26

CURSor Group

:CURSor?	Queries all cursor measurement settings.	5-27
:CURSor:BAR?	Queries all bar graph display cursor measurement settings.	5-27
:CURSor:BAR:LINKage	Sets or queries the on/off status of the cursor position linkage on the bar graph display.	5-27
:CURSor:BAR:POSition<x>	Sets or queries the position of the specified cursor on the bar graph display.	5-27
:CURSor:BAR[:STATe]	Sets or queries the on/off status of the cursor display on the bar graph display.	5-27
:CURSor:BAR:{Y<x>|DY}?	Queries the measured value of the specified cursor on the bar graph display.	5-27
:CURSor:TRENd?	Queries all trend display cursor measurement settings.	5-27
:CURSor:TRENd:LINKage	Sets or queries the on/off status of the cursor position linkage on the trend display.	5-27
:CURSor:TRENd:POSition<x>	Sets or queries the position of the specified cursor on the trend display.	5-27
:CURSor:TRENd[:STATe]	Sets or queries the on/off status of the cursor display on the trend display.	5-27
:CURSor:TRENd:TRACe<x>	Sets or queries the target of the specified cursor on the trend display.	5-28
:CURSor:TRENd:{X<x>|Y<x>|DY}?	Queries the measured value of the specified cursor on the trend display.	5-28
:CURSor:WAVE?	Queries all waveform display cursor measurement settings.	5-28
:CURSor:WAVE:LINKage	Sets or queries the on/off status of the cursor position linkage on the waveform display.	5-28
:CURSor:WAVE:PATH	Sets or queries the cursor path on the waveform display.	5-28
:CURSor:WAVE:POSition<x>	Sets or queries the position of the specified cursor on the waveform display.	5-28
:CURSor:WAVE[:STATe]	Sets or queries the on/off status of the cursor display on the waveform display.	5-28
:CURSor:WAVE:TRACe<x>	Sets or queries the target of the specified cursor on the waveform display.	5-28
:CURSor:WAVE:{X<x>|DX|PERDt|Y<x>|DY}?	Queries the measured value of the specified cursor on the waveform display.	5-28

DISPlay Group

:DISPlay?	Queries all display settings.	5-29
:DISPlay:BAR?	Queries all bar graph display settings.	5-29
:DISPlay:BAR:FORMat	Sets or queries the bar graph display format.	5-29
:DISPlay:BAR:ITEM<x>?	Queries all the display settings of the specified bar graph.	5-29

Command	Explanation	Page
:DISPlay:BAR:ITEM<x>[:FUNCtion]	Sets or queries the function and element of the specified bar graph item.	5-29
:DISPlay:BAR:ITEM<x>:SCALing?	Queries all scaling settings for the specified bar graph.	5-29
:DISPlay:BAR:ITEM<x>:SCALing:MODE	Sets or queries the scaling mode of the specified bar graph.	5-29
:DISPlay:BAR:ITEM<x>:SCALing:VALue	Sets or queries the upper limit of the manual scaling of the specified bar graph.	5-29
:DISPlay:BAR:ITEM<x>:SCALing:VERTical	Sets or queries the vertical scaling mode of the specified bar graph.	5-30
:DISPlay:BAR:ITEM<x>:SCALing:XAXis	Sets or queries the position of the X axis of the specified bar graph.	5-30
:DISPlay:BAR:ORDer	Sets or queries the displayed starting and ending harmonic orders of the bar graphs.	5-30
:DISPlay:HSPeed?	Queries all high speed data capturing display settings.	5-30
:DISPlay:HSPeed:COLumn?	Queries all column settings of the high speed data capturing mode.	5-30
:DISPlay:HSPeed:COLumn:ITEM<x>	Sets or queries the specified column display item of the high speed data capturing mode.	5-30
:DISPlay:HSPeed:COLumn:NUMber	Sets or queries the number of columns of the high speed data capturing mode.	5-30
:DISPlay:HSPeed:COLumn:RESet	Resets the column display items to their default values on the high speed data capturing mode.	5-30
:DISPlay:HSPeed:FRAMe	Sets or queries the on/off status of the high speed data capturing mode's data section frame.	5-30
:DISPlay:HSPeed:PAGE	Sets or queries the displayed page of the high speed data capturing mode.	5-31
:DISPlay:HSPeed:POVer	Sets or queries the on/off status of the display of peak over-range information in high speed data capturing mode.	5-31
:DISPlay:INFOrmation?	Queries all setup parameter list display settings.	5-31
:DISPlay:INFOrmation:PAGE	Sets or queries the displayed page of the setup parameter list display.	5-31
:DISPlay:INFOrmation[:STATe]	Sets or queries the on/off status of the setup parameter list display.	5-31
:DISPlay:MODE	Sets or queries the display mode.	5-31
:DISPlay:NUMeric?	Queries all numeric display settings.	5-31
:DISPlay:NUMeric:CUSTom?	Queries all numeric display settings in custom display mode.	5-31
:DISPlay:NUMeric:CUSTom:FILE:CDIRectory	Changes the directory that files are loaded from or saved to for the numeric display in custom display mode.	5-31
:DISPlay:NUMeric:CUSTom:FILE:DRIVe	Sets the drive that files are loaded from or saved to for the numeric display in custom display mode.	5-32
:DISPlay:NUMeric:CUSTom:FILE:FREE?	Queries the amount of free space (in bytes) on the drive that files are loaded from or saved to for the numeric display in custom display mode.	5-32
:DISPlay:NUMeric:CUSTom:FILE:LOAD:ABORt	Aborts a file loading operation for the numeric display in custom display mode.	5-32
:DISPlay:NUMeric:CUSTom:FILE:LOAD:BMP	Loads the specified background file for the numeric display in custom display mode.	5-32
:DISPlay:NUMeric:CUSTom:FILE:LOAD:BOTH	Loads the specified display configuration and background files for the numeric display in custom display mode.	5-32
:DISPlay:NUMeric:CUSTom:FILE:LOAD:ITEM	Loads the specified display configuration file for the numeric display in custom display mode.	5-32
:DISPlay:NUMeric:CUSTom:FILE:PATH?	Queries the absolute path of the directory that files are loaded from or saved to for the numeric display in custom display mode.	5-32

Command	Explanation	Page
:DISPlay:NUMeric:CUSTom:FILE:SAVE:ANAMing	Sets or queries the automatic file name generation feature for saving display configuration files of the numeric display in custom display mode.	5-32
:DISPlay:NUMeric:CUSTom:FILE:SAVE:ITEM	Saves the specified display configuration file for the numeric display in custom display mode.	5-32
:DISPlay:NUMeric:CUSTom:ITEM<x>?	Queries all the settings of the specified display item of the numeric display in custom display mode.	5-32
:DISPlay:NUMeric:CUSTom:ITEM<x>:COLor	Sets or queries the font color of the specified display item of the numeric display in custom display mode.	5-33
:DISPlay:NUMeric:CUSTom:ITEM<x>[:FUNCtion]	Sets or queries the display item (numeric item or string) of the numeric display in custom display mode.	5-33
:DISPlay:NUMeric:CUSTom:ITEM<x>:POSition	Sets or queries the display position of the specified display item of the numeric display in custom display mode.	5-34
:DISPlay:NUMeric:CUSTom:ITEM<x>:SIZE	Sets or queries the font size of the specified display item of the numeric display in custom display mode.	5-34
:DISPlay:NUMeric:CUSTom:PAGE	Sets or queries the displayed page of the numeric display in custom display mode.	5-34
:DISPlay:NUMeric:CUSTom:PERPage	Sets or queries the number of items displayed per page of the numeric display in custom display mode.	5-34
:DISPlay:NUMeric:CUSTom:TOTal	Sets or queries the total number of display items of the numeric display in custom display mode.	5-34
:DISPlay:NUMeric:FRAMe	Sets or queries the on/off status of the numeric display's data section frame.	5-34
:DISPlay:NUMeric:NORMal?	Queries all numeric display settings.	5-34
:DISPlay:NUMeric[:NORMal]:ALL?	Queries all settings of the numeric display in All Items display mode.	5-34
:DISPlay:NUMeric[:NORMal]:ALL:COLumn?	Queries all column settings of the numeric display in All Items display mode.	5-35
:DISPlay:NUMeric[:NORMal]:ALL:COLumn:DAELem	Sets or queries the on/off status of the column display all feature of the numeric display in All Items display mode.	5-35
:DISPlay:NUMeric[:NORMal]:ALL:COLumn:SCRoll	Sets or queries the on/off status of column scrolling of the numeric display in All Items display mode.	5-35
:DISPlay:NUMeric[:NORMal]:ALL:CURSor	Sets or queries the cursor position on the numeric display in All Items display mode.	5-35
:DISPlay:NUMeric[:NORMal]:ALL:ORDer	Sets or queries the displayed harmonic order on the harmonic measurement function display page of the numeric display in All Items display mode.	5-35
:DISPlay:NUMeric[:NORMal]:ALL:PAGE	Sets or queries the displayed page of the numeric display in All Items display mode.	5-35
:DISPlay:NUMeric[:NORMal]:FORMat	Sets or queries the numeric display format.	5-36
:DISPlay:NUMeric[:NORMal]:LIST?	Queries all numeric display settings in the list display modes.	5-36
:DISPlay:NUMeric[:NORMal]:LIST:CURSor	Sets or queries the cursor position on the numeric display in the list display modes.	5-36
:DISPlay:NUMeric[:NORMal]:LIST:HEADer	Sets or queries the cursor position of the header section on the numeric display in the list display modes.	5-36
:DISPlay:NUMeric[:NORMal]:LIST:ITEM<x>	Sets or queries the specified display item (function and element) on the numeric display in the list display modes.	5-36
:DISPlay:NUMeric[:NORMal]:LIST:ORDer	Sets or queries the harmonic order cursor position of the data section on the numeric display in the list display modes.	5-37
:DISPlay:NUMeric[:NORMal]:MATRix?	Queries all numeric display settings in matrix display mode.	5-37
:DISPlay:NUMeric[:NORMal]:MATRix:COLumn?	Queries all column settings of the numeric display in matrix display mode.	5-37

Command	Explanation	Page
:DISPlay:NUMeric[:NORMal]:MATRix:COLumn:ITEM<x>	Sets or queries the specified column display item of the numeric display in matrix display mode.	5-37
:DISPlay:NUMeric[:NORMal]:MATRix:COLumn:NUMber	Sets or queries the number of columns of the numeric display in matrix display mode.	5-37
:DISPlay:NUMeric[:NORMal]:MATRix:COLumn:RESet	Resets the column display items to their default values on the numeric display in matrix display mode.	5-37
:DISPlay:NUMeric[:NORMal]:MATRix:CURSor	Sets or queries the cursor position on the numeric display in matrix display mode.	5-37
:DISPlay:NUMeric[:NORMal]:MATRix:ITEM<x>	Sets or queries the specified display item (function and harmonic order) on the numeric display in matrix display mode.	5-38
:DISPlay:NUMeric[:NORMal]:MATRix:PAGE	Sets or queries the displayed page of the numeric display in matrix display mode.	5-38
:DISPlay:NUMeric[:NORMal]:MATRix:PRESet	Presets the display order pattern of displayed items on the numeric display in matrix display mode.	5-38
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}?	Queries all numeric display settings in 4 Items, 8 Items, or 16 Items display mode.	5-38
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}:CURSor	Sets or queries the cursor position on the numeric display in 4 Items, 8 Items, or 16 Items display mode.	5-38
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}:ITEM<x>	Sets or queries the function, element, and harmonic order of the specified numeric display item in 4 Items, 8 Items, or 16 Items display mode.	5-39
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}:PAGE	Sets or queries the displayed page of the numeric display in 4 Items, 8 Items, or 16 Items display mode.	5-39
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}:PRESet	Presets the display order pattern of displayed items on the numeric display in 4 Items, 8 Items, or 16 Items display mode.	5-39
:DISPlay:TRENd?	Queries all trend display settings.	5-39
:DISPlay:TRENd:ALL	Collectively sets the on/off status of all trends.	5-39
:DISPlay:TRENd:CLEar	Clears all trends.	5-39
:DISPlay:TRENd:FORMat	Sets or queries the display format of all trends.	5-39
:DISPlay:TRENd:ITEM<x>?	Queries all settings for the specified trend.	5-39
:DISPlay:TRENd:ITEM<x>[:FUNCtion]	Sets or queries the function, element, and harmonic order of the specified trend item.	5-40
:DISPlay:TRENd:ITEM<x>:SCALing?	Queries all scaling settings for the specified trend.	5-40
:DISPlay:TRENd:ITEM<x>:SCALing:MODE	Sets or queries the scaling mode of the specified trend.	5-40
:DISPlay:TRENd:ITEM<x>:SCALing:VALue	Sets or queries the upper and lower limits of the manual scaling of the specified trend.	5-40
:DISPlay:TRENd:T<x>	Sets or queries the on/off status of the specified trend.	5-40
:DISPlay:TRENd:TDIV	Sets or queries the trend horizontal axis (T/div).	5-40
:DISPlay:VECTor?	Queries all vector display settings.	5-41
:DISPlay:VECTor:FORMat	Sets or queries the display format of all vectors.	5-41
:DISPlay:VECTor:ITEM<x>?	Queries all settings for the specified vector.	5-41
:DISPlay:VECTor:ITEM<x>:OBJect	Sets or queries the wiring unit that is displayed using the specified vector.	5-41
:DISPlay:VECTor:ITEM<x>:{UMAG|IMAG}	Sets or queries the voltage or current zoom factor for the vector display.	5-41
:DISPlay:VECTor:NUMeric	Sets or queries the on/off status of the numeric data display on the vector display.	5-41
:DISPlay:WAVE?	Queries all waveform display settings.	5-41

Command	Explanation	Page
:DISPlay:WAVE:ALL	Collectively sets the on/off status of all waveform displays.	5-41
:DISPlay:WAVE:FORMat	Sets or queries the display format of all waveforms.	5-41
:DISPlay:WAVE:GRATicule	Sets or queries the graticule (grid) type.	5-41
:DISPlay:WAVE:INTerpolate	Sets or queries the waveform interpolation method.	5-41
:DISPlay:WAVE:MAPPing?	Queries all split screen waveform mapping settings.	5-41
:DISPlay:WAVE:MAPPing[:MODE]	Sets or queries the split screen waveform mapping mode.	5-41
:DISPlay:WAVE:MAPPing:{U<x>|I<x>|SPEed|TORQue|AUX<x>}	Sets or queries the split screen voltage, current, rotating speed, torque, or auxiliary signal waveform mapping setting.	5-42
:DISPlay:WAVE:POSition?	Queries all waveform vertical position (center position level) settings.	5-42
:DISPlay:WAVE:POSition:{U<x>|I<x>}	Sets or queries the vertical position (center position level) of the specified element's voltage or current waveform.	5-42
:DISPlay:WAVE:POSition:{UALL|IALL}	Collectively sets the vertical positions (center position levels) of the voltage or current waveforms of all elements.	5-42
:DISPlay:WAVE:SVALue	Sets or queries the on/off status of the scale value display.	5-42
:DISPlay:WAVE:TDIV	Sets or queries the waveform Time/div value.	5-42
:DISPlay:WAVE:TLABel	Sets or queries the on/off status of the waveform labels.	5-42
:DISPlay:WAVE:TRIGger?	Queries all trigger settings.	5-42
:DISPlay:WAVE:TRIGger:LEVel	Sets or queries the trigger level.	5-42
:DISPlay:WAVE:TRIGger:MODE	Sets or queries the trigger mode.	5-42
:DISPlay:WAVE:TRIGger:SLOPe	Sets or queries the trigger slope.	5-42
:DISPlay:WAVE:TRIGger:SOURce	Sets or queries the trigger source.	5-43
:DISPlay:WAVE:{U<x>|I<x>|SPEed|TORQue|AUX<x>}	Sets or queries the on/off status of the voltage, current, rotating speed, torque, or auxiliary signal waveform display.	5-43
:DISPlay:WAVE:VZoom?	Queries all waveform vertical zoom factor settings.	5-43
:DISPlay:WAVE:VZoom:{U<x>|I<x>}	Sets or queries the vertical zoom factor of the specified element's voltage or current waveform.	5-43
:DISPlay:WAVE:VZoom:{UALL|IALL}	Collectively sets the vertical zoom factor for the voltage or current waveforms of all elements.	5-43

FILE Group

:FILE?	Queries all file operation settings.	5-48
:FILE:CDIRectory	Changes the current directory.	5-48
:FILE:DELete:IMAGe:{BMP|PNG|JPEG}	Deletes the specified screen image data file.	5-48
:FILE:DELete:NUMeric:ASCii	Deletes the specified numeric data file.	5-48
:FILE:DELete:SETup	Deletes the specified setup parameter file.	5-48
:FILE:DELete:STORe:{DATA|HEADer}	Deletes the specified stored numeric data file.	5-48
:FILE:DELete:WAVE:ASCii	Deletes the specified waveform display data file.	5-48
:FILE:DRIVe	Sets the current drive.	5-48
:FILE:FILTer	Sets or queries the file list filter.	5-48
:FILE:FREE?	Queries the free space (in bytes) on the current drive.	5-48
:FILE:LOAD:ABORt	Aborts a file loading operation.	5-48
:FILE:LOAD:SETup	Loads the specified setup parameter file.	5-48
:FILE:PATH?	Queries the absolute path of the current directory.	5-49
:FILE:SAVE?	Queries all file save settings.	5-49
:FILE:SAVE:ABORt	Aborts a file saving operation.	5-49
:FILE:SAVE:ANAMing	Sets or queries the auto naming feature for saving files.	5-49

Command	Explanation	Page
:FILE:SAVE:COMMent	Sets or queries the comment that will be added to files that are saved.	5-49
:FILE:SAVE:NUMeric[:EXECute]	Saves numeric data to a file.	5-49
:FILE:SAVE:NUMeric:ITEM	Sets or queries the method that is used to select which items are saved when numeric data is saved to a file.	5-49
:FILE:SAVE:NUMeric:NORMal?	Queries all numeric data file save settings (for the manual save item selection method).	5-49
:FILE:SAVE:NUMeric:NORMal:ALL	Collectively sets the on/off status of the output of all element functions when numeric data is saved to a file.	5-49
:FILE:SAVE:NUMeric:NORMal:{ELEMent<x>|SIGMA|SIGMB|SIGMC}	Sets or queries the on/off status of the output of the specified element or wiring unit ΣA, ΣB, or ΣC when numeric data is saved to a file.	5-49
:FILE:SAVE:NUMeric:NORMal:<Function>	Sets or queries the on/off status of the specified function's output when numeric data is saved to a file.	5-50
:FILE:SAVE:NUMeric:NORMal:PRESet<x>	Presets the output on/off pattern of the element functions to be used when numeric data is saved to a file.	5-50
:FILE:SAVE:SETup[:EXECute]	Saves setup parameters to a file.	5-50
:FILE:SAVE:WAVE[:EXECute]	Saves waveform display data to a file.	5-50

HARMonics Group

:HARMonics<x>?	Queries all harmonic measurement settings.	5-51
:HARMonics<x>:CONFigure?	Queries the harmonic measurement groups of all elements.	5-51
:HARMonics<x>:CONFigure[:ALL]	Collectively sets the harmonic measurement group of all elements.	5-51
:HARMonics<x>:CONFigure:ELEMent<x>	Sets or queries the harmonic measurement group of the specified element.	5-51
:HARMonics<x>:CONFigure:{SIGMA|SIGMB|SIGMC}	Collectively sets the harmonic measurement group of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).	5-51
:HARMonics<x>:ORDer	Sets or queries the maximum and minimum harmonic orders that are analyzed.	5-51
:HARMonics<x>:PLLSource	Sets or queries the PLL source.	5-51
:HARMonics<x>:THD	Sets or queries the equation used to compute the THD (total harmonic distortion).	5-52
:HARMonics<x>:POINt	Sets or queries the number of FFT points to use for harmonic measurement.	5-52

HISTory Group

:HISTory?	Queries all history feature settings.	5-53
:HISTory:COMMunicate?	Queries all history communication output settings.	5-53
:HISTory:COMMunicate:COUNt	Sets or queries the history communication output count.	5-53
:HISTory:COMMunicate:FORMat	Sets or queries the history communication output data format.	5-53
:HISTory:COMMunicate:HOLD	Sets or queries the on/off (hold/release) status of the history communication output data hold feature.	5-53
:HISTory:COMMunicate:OUTPutmode	Sets or queries the history communication output mode.	5-53
:HISTory:NUMeric:LIST:VALue?	Queries the history communication numeric list data.	5-54
:HISTory:NUMeric:VALue?	Queries the history communication numeric data.	5-54

Command	Explanation	Page
HOLD Group		
:HOLD	Sets or queries the on/off status of the output hold feature for display, communication, and other types of data.	5-55
HSPeed Group		
:HSPeed?	Queries all high speed data capturing feature settings.	5-56
:HSPeed:CAPTured?	Queries the number of captures that have been performed in high speed data capturing.	5-56
:HSPeed:COUNt	Sets or queries the number of data captures.	5-56
:HSPeed:DISPlay?	Queries all display settings of high speed data capturing mode.	5-56
:HSPeed:DISPlay:COLumn?	Queries all column settings of high speed data capturing mode.	5-56
:HSPeed:DISPlay:COLumn:ITEM<x>	Sets or queries a column display item of high speed data capturing mode.	5-56
:HSPeed:DISPlay:COLumn:NUMber	Sets or queries the number of display columns of high speed data capturing mode.	5-56
:HSPeed:DISPlay:COLumn:RESet	Resets the column display items of high speed data capturing mode to their default values.	5-56
:HSPeed:DISPlay:FRAMe	Sets or queries the on/off status of the display's data section frame in high speed data capturing mode.	5-56
:HSPeed:DISPlay:PAGE	Sets or queries the display page of high speed data capturing mode.	5-56
:HSPeed:DISPlay:POVer	Sets or queries the on/off status of the display of peak over-range information in high speed data capturing mode.	5-56
:HSPeed:EXTSync	Sets or queries the on/off status of the high speed data capturing's external synchronization signal.	5-57
:HSPeed:FILTer?	Queries all high speed data capturing filter settings.	5-57
:HSPeed:FILTer[:HS]	Sets or queries the high speed data capturing digital filter (HS Filter).	5-57
:HSPeed:FILTer:LINE?	Queries all high speed data capturing line filter settings.	5-57
:HSPeed:FILTer:LINE[:ALL]	Sets the line filters of all the high speed data capturing elements.	5-57
:HSPeed:FILTer:LINE:ELEMent<x>	Sets or queries the line filter of the specified high speed data capturing element.	5-57
:HSPeed:MAXCount?	Sets or queries the maximum number of data captures.	5-57
:HSPeed:MEASuring?	Queries all high speed data capturing voltage mode or current mode settings.	5-57
:HSPeed:MEASuring[:ALL]	Sets all voltage and current modes at the same time.	5-57
:HSPeed:MEASuring:{U<x>|I<x>}	Sets or queries the specified voltage or current mode.	5-57
:HSPeed:MEASuring:{UALL|IALL}	Sets all voltage or current modes at the same time.	5-57
:HSPeed:POVer?	Queries the high speed data capturing peak over-range information.	5-58
:HSPeed:RECord?	Queries all high speed data capturing settings for saving data to files.	5-58
:HSPeed:RECord:FILE?	Queries all settings related to the saving of acquired data to files.	5-58
:HSPeed:RECord:FILE:ANAMing	Sets or queries the auto naming feature for saving acquired numeric data to files.	5-58
:HSPeed:RECord:FILE:CDIRectory	Changes the directory that acquired numeric data will be saved to.	5-58

Command	Explanation	Page
:HSPeed:RECord:FILE:CONVert?	Queries all settings related to the conversion of files of acquired numeric data into CSV format.	5-58
:HSPeed:RECord:FILE:CONVert:ABORt	Aborts the conversion of the specified file of acquired numeric data to CSV format.	5-58
:HSPeed:RECord:FILE:CONVert:AUTO	Sets or queries the on/off status of the automatic conversion of files of acquired numeric data to CSV format.	5-58
HSPeed:RECord:FILE:CONVert:CDIRectory	Changes the directory that acquired numeric data (CSV file) will be saved to.	5-58
:HSPeed:RECord:FILE:CONVert:DRIVe	Sets the drive that acquired numeric data (CSV file) is saved to.	5-58
:HSPeed:RECord:FILE:CONVert:EXECute	Converts the specified file of acquired numeric data to CSV format.	5-59
:HSPeed:RECord:FILE:CONVert:FREE?	Queries the free space (in bytes) on the drive that the acquired numeric data (CSV file) will be saved to.	5-59
:HSPeed:RECord:FILE:CONVert:PATH?	Queries the absolute path of the directory that the acquired numeric data (CSV file) will be saved to.	5-59
:HSPeed:RECord:FILE:CONVert:SPMode	Sets or queries the on/off state of the save location path designation mode of the acquired numeric data (CSV Path Selection Mode)	5-59
:HSPeed:RECord:FILE:DRIVe	Sets the drive that acquired numeric data is saved to.	5-59
:HSPeed:RECord:FILE:FREE?	Queries the free space (in bytes) on the drive that the acquired numeric data will be saved to.	5-59
:HSPeed:RECord:FILE:NAME	Sets or queries the name of the file that acquired numeric data will be saved to.	5-59
:HSPeed:RECord:FILE:PATH?	Queries the absolute path of the directory that the acquired numeric data will be saved to.	5-59
:HSPeed:RECord:FILE:STATe?	Queries the status of the file save operation being performed on the acquired numeric data.	5-59
:HSPeed:RECord:ITEM?	Queries all settings for the numeric data items that will be saved to a file.	5-59
:HSPeed:RECord:ITEM:AUX<x>	Sets or queries whether numeric data (auxiliary input) is saved to a file.	5-59
:HSPeed:RECord:ITEM:{I<x>|IA|IB|IC}	Sets or queries whether the specified element or wiring unit (ΣA, ΣB, or ΣC) of the numeric data (current) will be saved.	5-60
:HSPeed:RECord:ITEM:{P<x>|PA|PB|PC}	Sets or queries whether the specified element or wiring unit (ΣA, ΣB, or ΣC) of the numeric data (active power) will be saved.	5-60
:HSPeed:RECord:ITEM:{SPEed|TORQue|PM}	Sets or queries whether the rotating speed, torque, or motor output of the numeric data (motor) will be saved.	5-60
:HSPeed:RECord:ITEM:{U<x>|UA|UB|UC}	Sets or queries whether the specified element or wiring unit (ΣA, ΣB, or ΣC) of the numeric data (voltage) will be saved.	5-60
:HSPeed:RECord:ITEM:PRESet:ALL	Sets, at the same time, whether all numeric data items will be saved.	5-60
:HSPeed:RECord:ITEM:PRESet:{ELEMent<x>|SIGMA|SIGMB|SIGMC}	Sets, at the same time, whether the specified element or wiring unit (ΣA, ΣB, or ΣC) of all the types of numeric data will be saved.	5-60
:HSPeed:RECord:ITEM:PRESet:{U|I|P|MOTor|AUX}	Sets, at the same time, whether the specified functions of all types of numeric data will be saved.	5-60
:HSPeed:RECord[:STATe]	Sets or queries whether acquired numeric data is saved to a file.	5-60
:HSPeed:STARt	Starts data capturing.	5-60
:HSPeed:STATe?	Queries the status of high speed data capturing.	5-61
:HSPeed:STOP	Stops data capturing.	5-61
:HSPeed:TRIGger?	Queries all high speed data capturing trigger settings.	5-61

Command	Explanation	Page
:HSPeed:TRIGger:LEVel	Sets or queries the trigger level.	5-61
:HSPeed:TRIGger:MODE	Sets or queries the trigger mode.	5-61
:HSPeed:TRIGger:SLOPe	Sets or queries the trigger slope.	5-61
:HSPeed:TRIGger:SOURce	Sets or queries the trigger source.	5-61

IMAGe Group

:IMAGe?	Queries all screen image data output settings.	5-62
:IMAGe:ABORt	Aborts a screen image data output operation.	5-62
:IMAGe:COLor	Sets or queries the color tone of the screen image data that will be saved.	5-62
:IMAGe:COMMent	Sets or queries the comment displayed at the bottom of the screen.	5-62
:IMAGe:EXECute	Executes a screen image data output operation.	5-62
:IMAGe:FORMat	Sets or queries the format of the screen image data that will be saved.	5-62
:IMAGe:SAVE?	Queries all screen image data save settings.	5-62
:IMAGe:SAVE:ANAMing	Sets or queries the auto naming feature for saving files.	5-62
:IMAGe:SAVE:CDIRectory	Changes the directory that screen image data is saved to.	5-62
:IMAGe:SAVE:DRIVe	Sets the drive that screen image data is saved to.	5-62
:IMAGe:SAVE:FREE?	Queries the free space (in bytes) on the drive that the screen image data is saved to.	5-62
:IMAGe:SAVE:NAME	Sets or queries the name of the file that will be saved.	5-63
:IMAGe:SAVE:PATH?	Queries the absolute path of the directory that the screen image data is saved to.	5-63
:IMAGe:SEND?	Queries the screen image data.	5-63

INPut Group

:INPut?	Queries all input element settings.	5-64
[:INPut]:CFACtor	Sets or queries the crest factor.	5-64
[:INPut]:CURRent?	Queries all electric current measurement settings.	5-64
[:INPut]:CURRent:AUTO?	Queries the electric current auto range on/off statuses of all elements.	5-64
[:INPut]:CURRent:AUTO[:ALL]	Collectively sets the electric current auto range on/off status of all elements.	5-64
[:INPut]:CURRent:AUTO:ELEMent<x>	Sets or queries the electric current auto range on/off status of the specified element.	5-64
[:INPut]:CURRent:AUTO:{SIGMA|SIGMB|SIGMC}	Collectively sets the electric current auto range on/off status of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).	5-64
[:INPut]:CURRent:CONFig?	Queries the valid electric current ranges of all elements.	5-64
[:INPut]:CURRent:CONFig[:ALL]	Collectively sets the valid electric current range of all elements.	5-64
[:INPut]:CURRent:CONFig:ELEMent<x>	Sets or queries the valid electric current range of the specified element.	5-65
[:INPut]:CURRent:EXTSensor?	Queries all external current sensor range settings.	5-65
[:INPut]:CURRent:EXTSensor:CONFig?	Queries the valid external current sensor ranges of all elements.	5-65
[:INPut]:CURRent:EXTSensor:CONFig[:ALL]	Collectively sets the valid external current sensor range of all elements.	5-65

Command	Explanation	Page
[:INPut]:CURRent:EXTSensor:CONFig:ELEMent<x>	Sets or queries the valid external current sensor ranges of the specified element.	5-65
[:INPut]:CURRent:EXTSensor:DISPlay	Sets or queries the display mode of the external current sensor range.	5-65
[:INPut]:CURRent:EXTSensor:POJump?	Queries the jump destination ranges of all elements that are used when a current peak over-range occurs.	5-65
[:INPut]:CURRent:EXTSensor:POJump[:ALL]	Collectively sets the jump destination range of all elements that is used when a current peak over-range occurs.	5-66
[:INPut]:CURRent:EXTSensor:POJump:ELEMent<x>	Sets or queries the jump destination range of the specified element that is used when a current peak over-range occurs.	5-66
[:INPut]:CURRent:POJump?	Queries the jump destination ranges of all elements that are used when a current peak over-range occurs.	5-66
[:INPut]:CURRent:POJump[:ALL]	Collectively sets the jump destination range of all elements that is used when a current peak over-range occurs.	5-66
[:INPut]:CURRent:POJump:ELEMent<x>	Sets or queries the jump destination range of the specified element that is used when a current peak over-range occurs.	5-66
[:INPut]:CURRent:RANGe?	Queries the electric current ranges of all elements.	5-66
[:INPut]:CURRent:RANGe[:ALL]	Collectively sets the electric current range of all elements.	5-67
[:INPut]:CURRent:RANGe:ELEMent<x>	Sets or queries the electric current range of the specified element.	5-67
[:INPut]:CURRent:RANGe:{SIGMA|SIGMB|SIGMC}	Collectively sets the electric current range of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).	5-67
[:INPut]:CURRent:SPReset?	Queries the external current sensor conversion ratio presets of all elements.	5-67
[:INPut]:CURRent:SPReset[:ALL]	Collectively sets the external current sensor conversion ratio presets of all elements.	5-67
[:INPut]:CURRent:SPReset:ELEMent<x>	Sets or queries the external current sensor conversion ratio preset of the specified element.	5-67
[:INPut]:CURRent:SPReset:{SIGMA|SIGMB|SIGMC}	Collectively sets the external current sensor conversion ratio presets of all the elements that belong to the specified wiring unit (ΣA, ΣB or ΣC).	5-68
[:INPut]:CURRent:SRATio?	Queries the external current sensor conversion ratios of all elements.	5-68
[:INPut]:CURRent:SRATio[:ALL]	Collectively sets the external current sensor conversion ratios of all elements.	5-68
[:INPut]:CURRent:SRATio:ELEMent<x>	Sets or queries the external current sensor conversion ratio of the specified element.	5-68
[:INPut]:CURRent:SRATio:{SIGMA|SIGMB|SIGMC}	Collectively sets the external current sensor conversion ratios of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).	5-68
[:INPut]:ESELect	Sets or queries the element whose measurement range will be set.	5-68
[:INPut]:FILTer?	Queries all input filter settings.	5-68
[:INPut]:FILTer:FAUTo?	Queries the frequency filter (for when the data update interval is Auto) of all elements.	5-68
[:INPut]:FILTer:FAUTo[:ALL]	Collectively sets the frequency filter (for when the data update interval is Auto) of all elements.	5-68
[:INPut]:FILTer:FAUTo:ELEMent<x>	Sets or queries the frequency filter (for when the data update interval is Auto) of the specified element.	5-68
[:INPut]:FILTer:FREQuency?	Queries the frequency filters of all elements.	5-69
[:INPut]:FILTer:FREQuency[:ALL]	Collectively sets the frequency filter of all elements.	5-69
[:INPut]:FILTer:FREQuency:ELEMent<x>	Sets or queries the frequency filter of the specified element.	5-69

Command	Explanation	Page
[:INPut]:FILTer:LINE?	Queries the line filters of all elements.	5-69
[:INPut]:FILTer[:LINE][:ALL]	Collectively sets the line filter of all elements.	5-69
[:INPut]:FILTer[:LINE]:ELEMent<x>	Sets or queries the line filter of the specified element.	5-69
[:INPut]:FILTer[:LINE]:{SIGMA|SIGMB|SIGMC}	Collectively sets the line filter of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).	5-69
[:INPut]:INDependent	Sets or queries the on/off status of independent input element configuration.	5-69
[:INPut]:MODUle?	Queries the input element type.	5-70
[:INPut]:NULL:CONDition:{SPEed|TORQue|AUX<x>}	Queries the status of the NULL operation of rotating speed, torque, or AUX.	5-70
[:INPut]:NULL:CONDition:{U<x>|I<x>}	Queries the status of the voltage or current NULL operation of the specified element.	5-70
[:INPut]:NULL[:STATe]	Sets or queries the on/off status of the NULL feature.	5-70
[:INPut]:NULL:TARGet?	Queries all settings for the target of the NULL feature.	5-70
[:INPut]:NULL:TARGet[:MODE]	Sets or queries the selection mode for the target of the NULL feature.	5-70
[:INPut]:NULL:TARGet:{SPEed|TORQue|AUX<x>}	Sets or queries the target of the NULL operation (rotating speed, torque, or AUX).	5-70
[:INPut]:NULL:TARGet:{U<x>|I<x>}	Sets or queries the target of the voltage or current NULL operation of the specified element.	5-70
[:INPut]:NULL:TARGet:{UALL|IALL}	Collectively sets the target of the voltage or current NULL operation of all elements.	5-71
[:INPut]:POVer?	Queries the peak over-range information.	5-71
[:INPut]:SCALing?	Queries all scaling settings.	5-71
[:INPut]:SCALing:CTPReset?	Queries the CT ratio presets of all elements.	5-71
[:INPut]:SCALing:CTPReset[:ALL]	Collectively sets the CT ratio presets of all elements.	5-71
[:INPut]:SCALing:CTPReset:ELEMent<x>	Sets or queries the CT ratio preset of the specified element.	5-71
[:INPut]:SCALing:CTPReset:{SIGMA|SIGMB|SIGMC}	Collectively sets the CT ratio presets of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣB).	5-71
[:INPut]:SCALing:STATe?	Queries the on/off statuses of the scaling of all elements.	5-71
[:INPut]:SCALing[:STATe][:ALL]	Collectively sets the on/off status of the scaling of all elements.	5-71
[:INPut]:SCALing[:STATe]:ELEMent<x>	Sets or queries the on/off status of the scaling of the specified element.	5-71
[:INPut]:SCALing:{VT|CT|SFACtor}?	Queries the VT ratios, CT ratios, or power coefficients of all elements.	5-72
[:INPut]:SCALing:{VT|CT|SFACtor}[:ALL]	Collectively sets the VT ratio, CT ratio, or power coefficient of all elements.	5-72
[:INPut]:SCALing:{VT|CT|SFACtor}:ELEMent<x>	Sets or queries the VT ratio, CT ratio, or power coefficient of the specified element.	5-72
[:INPut]:SCALing:{VT|CT|SFACtor}:{SIGMA|SIGMB|SIGMC}	Collectively sets the VT ratio, CT ratio, or power coefficient of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).	5-72
[:INPut]:SYNChronize?	Queries the synchronization sources of all elements.	5-72
[:INPut]:SYNChronize[:ALL]	Collectively sets the synchronization source of all elements.	5-72
[:INPut]:SYNChronize:ELEMent<x>	Sets or queries the synchronization source of the specified element.	5-72
[:INPut]:SYNChronize:{SIGMA|SIGMB|SIGMC}	Collectively sets the synchronization source of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).	5-72
[:INPut]:SYNChronize:LEVel?	Queries all synchronization source level settings.	5-72

Command	Explanation	Page
[:INPut]:SYNChronize:LEVel:{VOLTage|CURRent|EXTSensor}[:ALL]	Sets the synchronization source level of the {voltage|current|external current sensor} of all elements at once.	5-72
[:INPut]:SYNChronize:LEVel:{VOLTage|CURRent|EXTSensor}:ELEMent<x>	Sets or queries the synchronization source level of the {voltage|current|external current sensor} of the specified element.	5-73
[:INPut]:SYNChronize:RECTifier?	Queries all synchronization source rectifier on/off settings.	5-73
[:INPut]:SYNChronize:RECTifier:{VOLTage|CURRent|EXTSensor}[:ALL]	Sets the synchronization source rectifier on/off setting of the {voltage|current|external current sensor} of all elements at once.	5-73
[:INPut]:SYNChronize:RECTifier:{VOLTage|CURRent|EXTSensor}:ELEMent<x>	Sets or queries the synchronization source rectifier on/off setting of the {voltage|current|external current sensor} of the specified element.	5-73
[:INPut]:VOLTage?	Queries all voltage measurement settings.	5-73
[:INPut]:VOLTage:AUTO?	Queries the voltage auto range on/off statuses of all elements.	5-73
[:INPut]:VOLTage:AUTO[:ALL]	Collectively sets the voltage auto range on/off status of all elements.	5-73
[:INPut]:VOLTage:AUTO:ELEMent<x>	Sets or queries the voltage auto range on/off status of the specified element.	5-73
[:INPut]:VOLTage:AUTO:{SIGMA|SIGMB|SIGMC}	Collectively sets the voltage auto range on/off status of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).	5-73
[:INPut]:VOLTage:CONFig?	Queries the valid voltage ranges of all elements.	5-73
[:INPut]:VOLTage:CONFig[:ALL]	Collectively sets the valid voltage range of all elements.	5-74
[:INPut]:VOLTage:CONFig:ELEMent<x>	Sets or queries the valid voltage ranges of the specified element.	5-74
[:INPut]:VOLTage:POJump?	Queries the jump destination ranges of all elements that are used when a voltage peak over-range occurs.	5-74
[:INPut]:VOLTage:POJump[:ALL]	Collectively sets the jump destination range of all elements that is used when a voltage peak over-range occurs.	5-74
[:INPut]:VOLTage:POJump:ELEMent<x>	Sets or queries the jump destination range of the specified element that is used when a voltage peak over-range occurs.	5-74
[:INPut]:VOLTage:RANGe?	Queries the voltage ranges of all elements.	5-74
[:INPut]:VOLTage:RANGe[:ALL]	Collectively sets the voltage range of all elements.	5-74
[:INPut]:VOLTage:RANGe:ELEMent<x>	Sets or queries the voltage range of the specified element.	5-74
[:INPut]:VOLTage:RANGe:{SIGMA|SIGMB|SIGMC}	Collectively sets the voltage range of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).	5-75
[:INPut]:WIRing	Sets or queries the wiring system.	5-75

INTEGrate Group

:INTEGrate?	Queries all integration settings.	5-76
:INTEGrate:ACAL	Sets or queries the on/off status of integration auto calibration.	5-76
:INTEGrate:INDependent	Sets or queries the on/off status of independent element integration.	5-76
:INTEGrate:MODE	Sets or queries the integration mode.	5-76
:INTEGrate:QMODe?	Queries the electric current modes for electric current integration of all elements.	5-76
:INTEGrate:QMODe[:ALL]	Collectively sets the electric current mode for electric current integration of all elements.	5-76
:INTEGrate:QMODe:ELEMent<x>	Sets or queries the electric current mode for electric current integration of the specified element.	5-76

Command	Explanation	Page
:INTEGrate:RACTion	Sets or queries the integration resume action that is taken when the power recovers from a power failure while integration is in progress.	5-76
:INTEGrate:RESet	Resets the integrated value.	5-76
:INTEGrate:RTALl:{STARt|END}	Collectively sets the integration start or end time of all elements for real-time integration mode.	5-77
:INTEGrate:RTIMe<x>?	Queries the integration start and end times for real-time integration mode.	5-77
:INTEGrate:RTIMe<x>:{STARt|END}	Sets or queries the integration start or end time for real-time integration mode.	5-77
:INTEGrate:STARt	Starts integration.	5-77
:INTEGrate:STATe?	Queries the integration status.	5-77
:INTEGrate:STOP	Stops integration.	5-78
:INTEGrate:TIMer<x>	Sets or queries the integration timer value.	5-78
:INTEGrate:TMALl	Collectively sets the integration timer of all elements.	5-78
:INTEGrate:WPTYpe?	Queries the watt-hour integration methods for each polarity (WP+/WP–) of all elements.	5-78
:INTEGrate:WPTYpe[:ALL]	Collectively sets the watt-hour integration method for each polarity (WP+/WP–) of all elements.	5-78
:INTEGrate:WPTYpe:ELEMent<x>	Sets or queries the watt-hour integration method for each polarity (WP+/WP–) of the specified element.	5-78

MEASure Group

:MEASure?	Queries all computation settings.	5-79
:MEASure:AVERaging?	Queries all averaging settings.	5-79
:MEASure:AVERaging:COUNt	Sets or queries the averaging coefficient.	5-79
:MEASure:AVERaging[:STATe]	Sets or queries the on/off status of averaging.	5-79
:MEASure:AVERaging:TYPE	Sets or queries the averaging type.	5-79
:MEASure:DMeasure?	Queries all delta computation settings.	5-79
:MEASure:DMeasure:MODE	Sets or queries the voltage or current mode that is used in delta computation.	5-79
:MEASure:DMeasure:{SIGMA|SIGMB|SIGMC}	Sets or queries the delta computation mode for wiring unit ΣA, ΣB, or ΣC.	5-79
:MEASure:EFFiciency?	Queries all efficiency computation settings.	5-80
:MEASure:EFFiciency:ETA<x>	Sets or queries the efficiency equation.	5-80
:MEASure:EFFiciency:UDEF<x>	Sets or queries the user-defined parameters used in the efficiency equation.	5-80
:MEASure:EVENt<x>?	Queries all the settings of the specified user-defined event.	5-80
:MEASure:EVENt<x>:EXPRession?	Queries all the settings of the specified user-defined event's expression.	5-80
:MEASure:EVENt<x>:EXPRession:CONDition	Sets or queries the specified user-defined event's expression (compound condition type).	5-80
:MEASure:EVENt<x>:EXPRession:INVerse	Sets or queries the on/off status of the logic inversion of the specified user-defined event's expression (compound condition type).	5-80
:MEASure:EVENt<x>:EXPRession:ITEM	Sets or queries the target item of the specified user-defined event's expression (range-defined type).	5-81
:MEASure:EVENt<x>:EXPRession:LIMit<x>	Sets or queries the range of the specified user-defined event's expression (range-defined type).	5-81
:MEASure:EVENt<x>:EXPRession:STRing?	Queries the specified user-defined event's expression as a string.	5-81

Command	Explanation	Page
:MEASure:EVENt<x>:EXPRession:TYPE	Sets or queries the specified user-defined event's expression type.	5-81
:MEASure:EVENt<x>:FLABel	Sets or queries the string that is displayed when the specified user-defined event's condition is not met.	5-81
:MEASure:EVENt<x>:NAME	Sets or queries the specified user-defined event's name.	5-81
:MEASure:EVENt<x>[:STATe]	Sets or queries the on/off status of the specified user-defined event.	5-81
:MEASure:EVENt<x>:TLABel	Sets or queries the string that is displayed when the specified user-defined event's condition is met.	5-82
:MEASure:FUNCtion<x>?	Queries all the settings of the specified user-defined function.	5-82
:MEASure:FUNCtion<x>:EXPRession	Sets or queries the equation of the specified user-defined function.	5-82
:MEASure:FUNCtion<x>:NAME	Sets or queries the name of the specified user-defined function.	5-82
:MEASure:FUNCtion<x>:PRESet[:EXECute]	Executes a user-defined function preset.	5-82
:MEASure:FUNCtion<x>:PRESet:FILE:SAVE	Saves the contents of user-defined functions to a file.	5-82
:MEASure:FUNCtion<x>:PRESet:FILE:LOAD	Loads a user-defined function file.	5-82
:MEASure:FUNCtion<x>[:STATe]	Sets or queries the on/off status of the specified user-defined function.	5-82
:MEASure:FUNCtion<x>:UNIT	Sets or queries the unit that is added to the computation result of the specified user-defined function.	5-82
:MEASure:MHOLd	Sets or queries the on/off status of the MAX HOLD feature used in user-defined functions.	5-83
:MEASure:PC?	Queries all Pc (corrected power) computation settings.	5-83
:MEASure:PC:IEC	Sets or queries the Pc (corrected power) equation.	5-83
:MEASure:PC:P<x>	Sets or queries a Pc (corrected power) equation parameter.	5-83
:MEASure:PHASe	Sets or queries the display format of the phase difference.	5-83
:MEASure:SAMPling	Sets or queries the sampling frequency.	5-83
:MEASure:SFORmula	Sets or queries the equation used to compute S (apparent power).	5-83
:MEASure:SQFormula	Sets or queries the equation used to compute S (apparent power) and Q (reactive power).	5-83
:MEASure:SYNChronize	Sets or queries the synchronized measurement mode.	5-84

MOTor Group

:MOTor?	Queries all motor evaluation function settings.	5-85
:MOTor:EANGle?	Queries all electrical angle measurement settings.	5-85
:MOTor:EANGle:CORRection?	Queries all electrical angle correction settings.	5-85
:MOTor:EANGle:CORRection:AENTer?	Queries all automatic electrical angle correction entry settings.	5-85
:MOTor:EANGle:CORRection:AENTer[:EXECute]	Executes an automatic electrical angle correction entry.	5-85
:MOTor:EANGle:CORRection:AENTer:TARGet	Sets or queries the target source for automatically entering the electrical angle correction value.	5-85
:MOTor:EANGle:CORRection:CLEar	Clears the electrical angle correction value.	5-85
:MOTor:EANGle:CORRection[:VALue]	Sets or queries the electrical angle correction value.	5-85
:MOTor:EANGle[:STATe]	Sets or queries the on/off status of electrical angle measurement.	5-85

Command	Explanation	Page
:MOTor:FILTer?	Queries all input filter settings.	5-85
:MOTor:FILTer[:LINE]	Sets or queries the line filter.	5-85
:MOTor:PM?	Queries all motor output (Pm) settings.	5-85
:MOTor:PM:SCALing	Sets or queries the motor output computation scaling factor.	5-86
:MOTor:PM:UNIT	Sets or queries the unit that is added to the motor output computation result.	5-86
:MOTor:POLE	Sets or queries the motor's number of poles.	5-86
:MOTor:SPEed?	Queries all rotating speed (Speed) settings.	5-86
:MOTor:SPEed:AUTO	Sets or queries the voltage auto range on/off status of the revolution signal (analog input type).	5-86
:MOTor:SPEed:LSCale?	Queries all revolution signal (analog input type) linear scaling settings.	5-86
:MOTor:SPEed:LSCale:AVALue	Sets or queries the slope (A) of the linear scale of the revolution signal (analog input type).	5-86
:MOTor:SPEed:LSCale:BVALue	Sets or queries the offset (B) of the linear scale of the revolution signal (analog input type).	5-86
:MOTor:SPEed:LSCale:CALCulate?	Queries all parameter calculation settings for the linear scale of the revolution signal (analog input type).	5-86
:MOTor:SPEed:LSCale:CALCulate:{P1X|P1Y|P2X|P2Y}	Sets or queries the data (Point1X, Point1Y, Point2X, or Point2Y) for parameter calculations of the linear scale of the revolution signal (analog input type).	5-86
:MOTor:SPEed:LSCale:CALCulate:EXECute	Calculates parameters for the linear scale of the revolution signal (analog input type).	5-86
:MOTor:SPEed:PRANge	Sets or queries the range of the revolution signal (pulse input type).	5-87
:MOTor:SPEed:PULSe	Sets or queries the number of pulses of the revolution signal (pulse input type).	5-87
:MOTor:SPEed:RANGe	Sets or queries the voltage range of the revolution signal (analog input type).	5-87
:MOTor:SPEed:SCALing	Sets or queries the rotating speed computation scaling factor.	5-87
:MOTor:SPEed:TYPE	Sets or queries the revolution signal input type.	5-87
:MOTor:SPEed:UNIT	Sets or queries the unit that is added to the rotating speed computation result.	5-87
:MOTor:SSPeed	Sets or queries the frequency measurement source for the synchronous speed (SyncSp) computation.	5-87
:MOTor:SYNChronize	Sets or queries the synchronization source for the rotating speed (Speed) and torque (Torque) computations.	5-87
:MOTor:TORQue?	Queries all torque (Torque) settings.	5-87
:MOTor:TORQue:AUTO	Sets or queries the voltage auto range on/off status of the torque signal (analog input type).	5-88
:MOTor:TORQue:LSCale?	Queries all torque signal (analog input type) linear scaling settings.	5-88
:MOTor:TORQue:LSCale:AVALue	Sets or queries the slope (A) of the linear scale of the torque signal (analog input type).	5-88
:MOTor:TORQue:LSCale:BVALue	Sets or queries the offset (B) of the linear scale of the torque signal (analog input type).	5-88
:MOTor:TORQue:LSCale:CALCulate?	Queries all parameter calculation settings for the linear scale of the torque signal (analog input type).	5-88
:MOTor:TORQue:LSCale:CALCulate:{P1X|P1Y|P2X|P2Y}	Sets or queries the data (Point1X, Point1Y, Point2X, or Point2Y) for parameter calculations of the linear scale of the torque signal (analog input type).	5-88

Command	Explanation	Page
:MOTor:TORQue:LSCale:CALCulate:EXECute	Calculates parameters for the linear scale of the torque signal (analog input type).	5-88
:MOTor:TORQue:PRANge	Sets or queries the range of the torque signal (pulse input type).	5-88
:MOTor:TORQue:RANGe	Sets or queries the voltage range of the torque signal (analog input type).	5-88
:MOTor:TORQue:RATE?	Queries all torque signal (pulse input type) rated-value settings.	5-89
:MOTor:TORQue:RATE:{UPPer|LOWer}	Sets or queries the upper or lower limit of the rated value of the torque signal (pulse input type).	5-89
:MOTor:TORQue:SCALing	Sets or queries the torque computation scaling factor.	5-89
:MOTor:TORQue:TYPE	Sets or queries the torque signal input type.	5-89
:MOTor:TORQue:UNIT	Sets or queries the unit that is added to the torque computation result.	5-89

NUMeric Group

:NUMeric?	Queries all numeric data output settings.	5-90
:NUMeric:BYTeorder	Sets or queries the output byte order of the numeric data (FLOAT format).	5-90
:NUMeric:FORMat	Sets or queries the numeric data format.	5-90
:NUMeric:HSPeed?	Queries all numeric data output settings of the high speed data capturing mode.	5-90
:NUMeric:HSPeed:CLEar	Clears high speed data capturing mode numeric list data output items (sets the items to NONE).	5-90
:NUMeric:HSPeed:DELete	Deletes high speed data capturing mode numeric list data output items.	5-90
:NUMeric:HSPeed:HEADer?	Queries the header of the numeric data of high speed data capturing mode.	5-91
:NUMeric:HSPeed:ITEM<x>	Sets or queries the output item (function and element) of the specified high speed data capturing mode numeric data item.	5-91
:NUMeric:HSPeed:{MAXimum|MINimum}?	Queries the maximum or minimum value of the numeric data of high speed data capturing mode.	5-91
:NUMeric:HSPeed:NUMber	Sets or queries the number of numeric data items that are transmitted by the :NUMeric:HSPeed:VALue? command.	5-91
:NUMeric:HSPeed:PRESet	Presets the numeric data output item pattern of the high speed data capturing mode.	5-92
:NUMeric:HSPeed:VALue?	Queries the numeric data of high speed data capturing mode.	5-92
:NUMeric:HOLD	Sets or queries the on/off (hold/release) status of the numeric data hold feature.	5-93
:NUMeric:LIST?	Queries all harmonic measurement numeric list data output settings.	5-93
:NUMeric:LIST:CLEar	Clears harmonic measurement numeric list data output items (sets the items to NONE).	5-93
:NUMeric:LIST:DELete	Deletes harmonic measurement numeric list data output items.	5-93
:NUMeric:LIST:ITEM<x>	Sets or queries the output item (function and element) of the specified harmonic measurement numeric list data item.	5-94
:NUMeric:LIST:NUMber	Sets or queries the number of numeric list data items that are transmitted by :NUMeric:LIST:VALue?.	5-94
:NUMeric:LIST:ORDer	Sets or queries the maximum output harmonic order of the harmonic measurement numeric list data.	5-94

Command	Explanation	Page
:NUMeric:LIST:PRESet	Presets the harmonic measurement numeric list data output item pattern.	5-94
:NUMeric:LIST:SELect	Sets or queries the output components of the harmonic measurement numeric list data.	5-94
:NUMeric:LIST:VALue?	Queries the harmonic measurement numeric list data.	5-95
:NUMeric:NORMal?	Queries all numeric data output settings.	5-95
:NUMeric[:NORMal]:CLEar	Clears numeric data output items (sets the items to NONE).	5-95
:NUMeric[:NORMal]:DELete	Deletes numeric data output items.	5-95
:NUMeric[:NORMal]:ITEM<x>	Sets or queries the specified numeric data output item (function, element, and harmonic order).	5-96
:NUMeric[:NORMal]:NUMber	Sets or queries the number of numeric data items that are transmitted by the :NUMeric[:NORMal]:VALue? command.	5-96
:NUMeric[:NORMal]:PRESet	Presets the numeric data output item pattern.	5-96
:NUMeric[:NORMal]:VALue?	Queries the numeric data.	5-96

RATE Group

:RATE[:RATE]	Sets or queries the data update interval.	5-102
:RATE:AUTO?	Queries all applicable settings for when the data update interval is set to Auto.	5-102
:RATE:AUTO:TIMeout	Sets or queries the timeout for when the data update interval is set to Auto.	5-102

STATus Group

:STATus?	Queries all the settings for the communication status feature.	5-103
:STATus:CONDition?	Queries the contents of the condition register.	5-103
:STATus:EESE	Sets or queries the extended event enable register.	5-103
:STATus:EESR?	Queries the contents of the extended event register and clears the register.	5-103
:STATus:ERRor?	Queries the error code and message of the last error that has occurred (top of the error queue).	5-103
:STATus:FILTer<x>	Sets or queries the transition filter.	5-103
:STATus:QENable	Sets or queries whether messages other than errors will be stored to the error queue (ON/OFF).	5-103
:STATus:QMESsage	Sets or queries whether message information will be attached to the response to the :STATus:ERRor? query (ON/OFF).	5-103
:STATus:SPOLl?	Executes serial polling.	5-103

STORe Group

:STORe?	Queries all numeric data storage settings.	5-104
:STORe:COUNt	Sets or queries the storage count.	5-104
:STORe:FILE?	Queries all settings related to the saving of the data stored in this instrument to files.	5-104
:STORe:FILE:ANAMing	Sets or queries the auto naming feature for saving stored numeric data to files.	5-104
:STORe:FILE:CDIRectory	Changes the directory that stored numeric data is saved to.	5-104
:STORe:FILE:CONVert?	Queries all settings related to the conversion of stored numeric data files into CSV format.	5-104
:STORe:FILE:CONVert:ABORt	Aborts the conversion of a numeric data file to CSV format.	5-104

Command	Explanation	Page
:STORe:FILE:CONVert:AUTO	Sets or queries the on/off status of the automatic conversion of stored numeric data files to CSV format.	5-104
:STORe:FILE:CONVert:CDIRectory	Changes the directory that stored numeric data (CSV file) is saved to.	5-104
:STORe:FILE:CONVert:DRIVe	Sets the drive that stored numeric data (CSV file) is saved to.	5-104
:STORe:FILE:CONVert:EXECute	Converts the specified stored numeric data file to CSV format.	5-104
:STORe:FILE:CONVert:FREE?	Queries the free space (in bytes) on the drive that the stored numeric data (CSV file) is saved to.	5-105
:STORe:FILE:CONVert:PATH?	Queries the absolute path of the directory that the stored numeric data (CSV file) is saved to.	5-105
:STORe:FILE:CONVert:SPMode	Sets or queries the on/off state of the save location path designation mode of the stored numeric data (CSV Path Selection Mode)	5-105
:STORe:FILE:DRIVe	Sets the drive that stored numeric data is saved to.	5-105
:STORe:FILE:FREE?	Queries the free space (in bytes) on the drive that the stored numeric data is saved to.	5-105
:STORe:FILE:NAME	Sets or queries the name of the file that stored numeric data is saved to.	5-105
:STORe:FILE:PATH?	Queries the absolute path of the directory that the stored numeric data is saved to.	5-105
:STORe:INTerval	Sets or queries the storage interval.	5-105
:STORe:NUMeric?	Queries all numeric data storage item settings.	5-105
:STORe:NUMeric:ITEM	Sets or queries the numeric data storage item selection method.	5-105
:STORe:NUMeric:NORMal?	Queries all numeric data storage item settings (for the manual selection method).	5-105
:STORe:NUMeric[:NORMal]:ALL	Collectively sets the on/off status of the output of all element functions when numeric data is stored.	5-106
:STORe:NUMeric[:NORMal]:{ELEMent<x>|SIGMA|SIGMB|SIGMC}	Sets or queries the on/off status of the output of the specified element or wiring unit ΣA, ΣB, or ΣC when numeric data is stored.	5-106
:STORe:NUMeric[:NORMal]:<Function>	Sets or queries the on/off status of the specified function's output when numeric data is stored.	5-106
:STORe:NUMeric[:NORMal]:PRESet<x>	Presets the output on/off pattern of the element functions to be used when numeric data is stored.	5-106
:STORe:RESet	Resets the numeric data storage feature.	5-106
:STORe:RTIMe?	Queries the storage start and end times for real-time storage mode.	5-106
:STORe:RTIMe:{STARt|END}	Sets or queries the storage start or end time for real-time storage mode.	5-106
:STORe:SASTart	Sets or queries whether numeric data is stored when storage starts.	5-106
:STORe:SMODe	Sets or queries the storage mode.	5-106
:STORe:STARt	Begins the storing of numeric data.	5-106
:STORe:STATe?	Sets or queries the storage state.	5-107
:STORe:STOP	Stops the storing of numeric data.	5-107
:STORe:TEVent	Sets or queries the event that the event-synchronized storage mode will trigger on.	5-107

Command	Explanation	Page
SYSTem Group		
:SYSTem?	Queries all system settings.	5-108
:SYSTem:CLOCk?	Queries all date/time settings.	5-108
:SYSTem:CLOCk:DISPlay	Sets or queries the on/off status of the date/time display.	5-108
:SYSTem:CLOCk:SNTP?	Queries all settings related to using SNTP to set the date and time.	5-108
:SYSTem:CLOCk:SNTP[:EXECute]	Uses SNTP to set the date and time.	5-108
:SYSTem:CLOCk:SNTP:GMTTime	Sets or queries the time difference from Greenwich Mean Time.	5-108
:SYSTem:CLOCk:TYPE	Sets or queries the date/time setup method.	5-108
:SYSTem:COMMunicate:COMMand	Sets or queries the communication command type.	5-108
:SYSTem:DATE	Sets or queries the date.	5-108
:SYSTem:DFLow:FREQuency	Sets or queries the frequency data display format when a low frequency (or no frequency) input is applied.	5-108
:SYSTem:DFLow:MOTor	Sets or queries the motor data display format when no pulse is applied.	5-108
:SYSTem:DPOint	Sets or queries the type of decimal point that is used when saving various data in ASCII format (CSV).	5-109
:SYSTem:ECLear	Clears error messages displayed on the screen.	5-109
:SYSTem:FONT	Sets or queries the menu and message font size.	5-109
:SYSTem:KLOCk	Sets or queries the on/off status of the key lock.	5-109
:SYSTem:LANGuage?	Queries all display language settings.	5-109
:SYSTem:LANGuage:MENU	Sets or queries the menu language.	5-109
:SYSTem:LANGuage:MESSage	Sets or queries the message language.	5-109
:SYSTem:LCD?	Queries all LCD settings.	5-109
:SYSTem:LCD:AOFF?	Queries all the settings for the feature that automatically turns off the backlight.	5-109
:SYSTem:LCD:AOFF[:STATe]	Sets or queries the on/off status of the feature that automatically turns off the backlight.	5-109
:SYSTem:LCD:AOFF:TIME	Sets or queries the amount of time until the backlight is automatically turned off.	5-109
:SYSTem:LCD:BRIGhtness	Sets or queries the LCD brightness.	5-109
:SYSTem:LCD:COLor?	Queries all LCD color settings.	5-109
:SYSTem:LCD:COLor:BASecolor	Sets or queries the screen (menu) base color.	5-109
:SYSTem:LCD:COLor:GRAPh?	Queries all waveform color settings.	5-110
:SYSTem:LCD:COLor:GRAPh:CHANnel<x>	Sets or queries the specified waveform's color.	5-110
:SYSTem:LCD:COLor:GRAPh:PRESet	Presets the waveform color pattern.	5-110
:SYSTem:LCD:COLor:INTENsity:GRID	Sets or queries the grid intensity.	5-110
:SYSTem:LCD[:STATe]	Sets or queries the on/off status of the backlight.	5-110
:SYSTem:MODel?	Queries the model code.	5-110
:SYSTem:RZERo	Sets or queries the on/off status of the rounding to zero feature.	5-110
:SYSTem:RESolution	Sets or queries the numeric data display resolution.	5-110
:SYSTem:SERial?	Queries the serial number.	5-110
:SYSTem:SUFFix?	Queries the suffix code.	5-110
:SYSTem:TIME	Sets or queries the time.	5-111
:SYSTem:USBKeyboard	Sets or queries the USB keyboard type.	5-111

Command	Explanation	Page
WAVeform Group		
:WAVeform?	Queries all waveform display data output settings.	5-112
:WAVeform:BYTeorder	Sets or queries the output byte order of the waveform display data (FLOAT format) that is transmitted by the :WAVeform:SEND? command.	5-112
:WAVeform:END	Sets or queries the output end point of the waveform display data that is transmitted by the :WAVeform:SEND? command.	5-112
:WAVeform:FORMat	Sets or queries the format of the waveform display data that is transmitted by the :WAVeform:SEND? command.	5-112
:WAVeform:HOLD	Sets or queries the on/off (hold/release) status of the waveform display data hold feature for all waveforms.	5-112
:WAVeform:LENGth?	Queries the total number of points of the waveform specified by the :WAVeform:TRACe command.	5-112
:WAVeform:SEND?	Queries the waveform display data specified by the :WAVeform:TRACe command.	5-113
:WAVeform:SRATe?	Queries the sample rate of the acquired waveform.	5-113
:WAVeform:STARt	Sets or queries the output start point of the waveform display data that is transmitted by the :WAVeform:SEND? command.	5-113
:WAVeform:TRACe	Sets or queries the target waveform for the :WAVeform:SEND? command.	5-113
:WAVeform:TRIGger?	Queries the trigger position of the acquired waveform.	5-113
Common Command Group		
*CAL?	Executes zero calibration (zero-level compensation, the same operation as pressing CAL—SHIFT+SINGLE) and queries the result.	5-114
*CLS	Clears the standard event register, extended event register, and error queue.	5-114
*ESE	Sets or queries the standard event enable register.	5-114
*ESR?	Queries and clears the standard event register.	5-114
*IDN?	Queries the instrument model.	5-114
*OPC	Sets bit 0 (the OPC bit) of the standard event register to 1 upon the completion of the specified overlap command.	5-114
*OPC?	Returns ASCII code 1 if the specified overlap command has finished.	5-115
*OPT?	Queries the installed options.	5-115
*RST	Initializes the settings.	5-115
*SRE	Sets or queries the service request enable register.	5-115
*STB?	Queries the status byte register.	5-115
*TRG	Executes single measurement (the same operation as when SINGLE is pressed).	5-115
*TST?	Performs a self-test and queries the result.	5-116
*WAI	Holds the execution of the subsequent command until the completion of the specified overlap command.	5-116

5.2 AOUTput Group

The commands in this group deal with D/A output.
You can make the same settings and queries that you can make by pressing UTILITY on the front panel, and then using the D/A Output Items menu.
The commands in this group are only valid on models with the D/A output (/DA) option.

:AOUTput?

Queries all D/A output settings.

Syntax :AOUTput?

:AOUTput:NORMal?

Queries all D/A output settings.

Syntax :AOUTput:NORMal?

:AOUTput[:NORMal]:CHANnel<x>

Sets or queries a D/A output item (function, element, or harmonic order).

Syntax :AOUTput[:NORMal]:CHANnel<x> {NONE|<Function>[,<Element>][,<Order>]}
:AOUTput[:NORMal]:CHANnel<x>?
<x> = 1 to 20 (output channel)
NONE = No output item
<Function> = {URMS|IRMS|P|S|Q|...}
<Element> = {<NRf>|SIGMa|SIGMB|SIGMC}
(<NRf> = 1 to 6)
<Order> = {TOTal|DC|<NRf>}
(<NRf> = 1 to 500)

Example
```
:AOUTPUT:NORMAL:CHANNEL1 URMS,1
:AOUTPUT:NORMAL:CHANNEL1?
-> :AOUTPUT:NORMAL:CHANNEL1 URMS,1
:AOUTPUT:NORMAL:CHANNEL1 UK,1,1
:AOUTPUT:NORMAL:CHANNEL1?
-> :AOUTPUT:NORMAL:CHANNEL1 UK,1,1
```

Description
- For information about the options available for <Function>, see “Numeric data functions” in “Function option list” on page 5-44.
- If <Element> is omitted, the element is set to 1.
- If <Order> is omitted, the order is set to TOTal.
- <Element> and <Order> are omitted from responses to functions that do not need them.

:AOUTput[:NORMal]:IRTime

Sets or queries the integration time that is used in the D/A output of the integrated value.

Syntax :AOUTput[:NORMal]:IRTime {<NRf>,<NRf>,<NRf>}
:AOUTput[:NORMal]:IRTime?
{<NRf>,<NRf>,<NRf>} = 0,0,0 to 10000,0,0
First <NRf> = 0 to 10000 (hours)
Second <NRf> = 0 to 59 (minutes)
Third <NRf> = 0 to 59 (seconds)

Example
```
:AOUTPUT:NORMAL:IRTIME 1,0,0
:AOUTPUT:NORMAL:IRTIME?
-> :AOUTPUT:NORMAL:IRTIME 1,0,0
```

:AOUTput[:NORMal]:MODE<x>

Sets or queries the rated-value setup mode for D/A output items.

Syntax :AOUTput[:NORMal]:MODE<x>{FIXed|MANual}
:AOUTput[:NORMal]:MODE<x>?
<x> = 1 to 20 (output channel)

Example
```
:AOUTPUT:NORMAL:MODE1 FIXED
:AOUTPUT:NORMAL:MODE1?
-> :AOUTPUT:NORMAL:MODE1 FIXED
```

:AOUTput[:NORMal]:RATE<x>

Sets or queries the rated maximum or minimum value for D/A output items.

Syntax :AOUTput[:NORMal]:RATE<x> {<NRf>,<NRf>}
:AOUTput[:NORMal]:RATE<x>?
<x> = 1 to 20 (output channel)
<NRf> = –9.999E+12 to 9.999E+12

Example
```
:AOUTPUT:NORMAL:RATE1 100,-100
:AOUTPUT:NORMAL:RATE1?
-> :AOUTPUT:NORMAL:RATE1 100.0E+00,
-100.0E+00
```

Description
- Set the upper limit and then the lower limit.
- This setting is valid when the D/A output rated-value setup mode (:AOUTput [:NORMal]: MODE<x>) is set to MANual.

5.3 AUX Group

The commands in this group deal with the auxiliary input feature.
You can make the same settings and queries that you can make by pressing MOTOR/AUX SET (SHIFT+SCALING) on the front panel.
The commands in this group are only valid on models with the auxiliary input (/AUX) option.

:AUX<x>?

Queries all auxiliary input settings.

Syntax :AUX<x>?
<x> = 1 or 2 (AUX input channel)

:AUX<x>:AUTO

Sets or queries the voltage auto range on/off status of the specified auxiliary input.

Syntax :AUX<x>:AUTO {<Boolean>}
:AUX<x>:AUTO?
<x> = 1 or 2 (AUX input channel)
Example :AUX1:AUTO ON
:AUX1:AUTO? -> :AUX1:AUTO 1

:AUX<x>:FILTer?

Queries all input filter settings for the auxiliary inputs.

Syntax :AUX<x>:FILTer?
Description The <x> value in AUX<x> has no meaning in the setting or query.

:AUX<x>:FILTer[:LINE]

Sets or queries the line filter for the auxiliary inputs.

Syntax :AUX<x>:FILTer[:LINE] {OFF|<Frequency>}
:AUX<x>:FILTer:LINE?
OFF = Line filter off
<Frequency> = 100 Hz, 1 kHz (when the line filter is on; cutoff frequency)
Example :AUX:FILTER:LINE OFF
:AUX:FILTER:LINE?
-> :AUX1:FILTER:LINE OFF
Description The <x> value in AUX<x> has no meaning in the setting or query.

:AUX<x>:LSCale?

Queries all auxiliary input linear scaling settings.

Syntax :AUX<x>:LSCale?
<x> = 1 or 2 (AUX input channel)

:AUX<x>:LSCale:AVALue

Sets or queries the slope (A) of the linear scale of the auxiliary input feature.

Syntax :AUX<x>:LSCale:AVALue {<NRf>}
:AUX<x>:LSCale:AVALue?
<x> = 1 or 2 (AUX input channel)
<NRf> = 1.000E-03 to 1.000E+06
Example :AUX1:LSCALE:AVALUE 1.000
:AUX1:LSCALE:AVALUE?
-> :AUX1:LSCALE:AVALUE 1.000E+00

:AUX<x>:LSCale:BVALue

Sets or queries the offset (B) of the linear scale of the auxiliary input feature.

Syntax :AUX<x>:LSCale:BVALue {<NRf>}
:AUX<x>:LSCale:BVALue?
<x> = 1 or 2 (AUX input channel)
<NRf> = -1.000E+06 to 1.000E+06
Example :AUX1:LSCALE:BVALUE 0
:AUX1:LSCALE:BVALUE?
-> :AUX1:LSCALE:BVALUE 0.000E+00

:AUX<x>:LSCale:CALCulate?

Queries all parameter calculation settings for the linear scale of the auxiliary input feature.

Syntax :AUX<x>:LSCale:CALCulate?
<x> = 1 or 2 (AUX input channel)

:AUX<x>:LSCale:CALCulate:{P1X|P1Y|P2X|P2Y}

Sets or queries the data (Point1X, Point1Y, Point2X, or Point2Y) for parameter calculations of the linear scale of the auxiliary input feature.

Syntax :AUX<x>:LSCale:CALCulate:{P1X|P1Y|P2X|P2Y} {<NRf>}
:AUX<x>:LSCale:CALCulate:{P1X|P1Y|P2X|P2Y}?
<x> = 1 or 2 (AUX input channel)
<NRf> = -1.000E+12 to 1.000E+12
Example :AUX1:LSCALE:CALCULATE:P1X 0
:AUX1:LSCALE:CALCULATE:P1X?
-> :AUX1:LSCALE:CALCULATE:
P1X 0.000E+00

:AUX<x>:LSCale:CALCulate:EXECute

Calculates parameters for the linear scale of the auxiliary input feature.

Syntax :AUX<x>:LSCale:CALCulate:EXECute
<x> = 1 or 2 (AUX input channel)

Example :AUX1:LSCALE:CALCULATE:EXECUTE

Description This instrument uses the data that has been specified (Point1X, Point1Y, Point2X, and Point2Y) to calculate and set the slope (A) and offset (B) of the linear scale.

:AUX<x>:NAME

Sets or queries the auxiliary input name.

Syntax :AUX<x>:NAME {<String>}
:AUX<x>:NAME?
<x> = 1 or 2 (AUX input channel)
<String> = Up to 8 characters

Example :AUX1:NAME "AUX1"
:AUX1:NAME?
-> :AUX1:NAME "AUX1"

:AUX<x>:RANGe

Sets or queries the auxiliary input voltage range.

Syntax :AUX<x>:RANGe {<Voltage>}
:AUX<x>:RANGe?
<x> = 1 or 2 (AUX input channel)
<Voltage> = 50 mV, 100 mV, 200 mV, 500 mV, 1 V, 2 V, 5 V, 10 V, 20 V

Example :AUX1:RANGE 20V
:AUX1:RANGE?
-> :AUX1:RANGE 20.00E+00

:AUX<x>:SCALing

Sets or queries the auxiliary input scaling factor.

Syntax :AUX<x>:SCALing {<NRf>}
:AUX<x>:SCALing?
<x> = 1 or 2 (AUX input channel)
<NRf> = 0.0001 to 99999.9999

Example :AUX1:SCALING 1
:AUX1:SCALING?
-> :AUX1:SCALING 1.0000

:AUX<x>:UNIT

Sets or queries the unit to assign to the auxiliary input.

Syntax :AUX<x>:UNIT {<String>}
:AUX<x>:UNIT?
<x> = 1 or 2 (AUX input channel)
<String> = Up to 8 characters

Example :AUX1:UNIT "kW/m2"
:AUX1:UNIT?
-> :AUX1:UNIT "kW/m2"

Description This command has no effect on the computation result.

5.4 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?

Queries all communication settings.

Syntax :COMMunicate?

:COMMunicate:HEADer

Sets or queries whether a header is added to the response to a query. (Example with header: ":DISPLAY:MODE NUMERIC." Example without header: "NUMERIC.")

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

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

:COMMunicate:LOCKout

Sets or clears local lockout.

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

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

:COMMunicate:OPSE

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:OPSR? command.

Example :COMMUNICATE:OPSE 65535
:COMMUNICATE:OPSE?
-> :COMMUNICATE:OPSE 64

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 bit 6.

:COMMunicate:OPSR?

Queries the operation pending status register.

Syntax :COMMunicate:OPSR?

Example :COMMunicate:OPSR? -> 0

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	0	0	0	0	0	ACS	0	0	0	0	0	0

When bit 6 (ACS) = 1:
Storage media access is incomplete.

:COMMunicate:OVERlap

Sets or queries the commands that operate as overlap commands.

Syntax :COMMunicate:OVERlap <Register>
:COMMunicate:OVERlap?
<Register> = 0 to 65535,
see the figure for the :COMMunicate:OPSR? command.

Example :COMMUNICATE:OVERLAP 65535
:COMMUNICATE:OVERLAP?
-> :COMMUNICATE:OVERLAP 64

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 bit 6.
- For information about how to synchronize a program using :COMMunicate:OVERlap, see page 4-12.
- In the above example, bit 6 is set to 1 to make all overlap commands applicable (see the figure for the :COMMunicate:OPSR? command).

:COMMunicate:REMote

Sets this 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

Sets or queries whether the response to a query is returned fully spelled out (example: ":INPUT:VOLTAGE:RANGE:ELEMENT1 1.000E+03") or in its abbreviated form (example: "VOLT:RANG:ELEM 1.000E+03").

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

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

:COMMunicate:WAIT

Waits for a specified extended event to occur.

Syntax `:COMMunicate:WAIT <Register>`
<Register> = 0 to 65535 (Extended event register. For more information, see page 6-7.)

Example `:COMMUNICATE:WAIT 1`

Description For information about how to synchronize a program using :COMMunicate:WAIT, see page 4-14.

:COMMunicate:WAIT?

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 page 6-7.)

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

5.5 CURSor Group

The commands in this group deal with cursor measurements. You can make the same settings and queries (of settings and measured values) that you can make by pressing CURSOR (SHIFT+FORM) on the front panel.

:CURSor?

Queries all cursor measurement settings.

Syntax :CURSor?

:CURSor:BAR?

Queries all bar graph display cursor measurement settings.

Syntax :CURSor:BAR?

Description The bar graph display's cursor feature is only available on models with the harmonic measurement (/G5 or /G6) option.

:CURSor:BAR:LINKage

Sets or queries the on/off status of the cursor position linkage on the bar graph display.

Syntax :CURSor:BAR:LINKage {<Boolean>}
:CURSor:BAR:LINKage?

Example :CURSOR:BAR:LINKAGE OFF
:CURSOR:BAR:LINKAGE?
-> :CURSOR:BAR:LINKAGE 0

:CURSor:BAR:POSition<x>

Sets or queries the position of the specified cursor on the bar graph display.

Syntax :CURSor:BAR:POSition<x> {<NRf>}
:CURSor:BAR:POSition<x>?
<x> = 1, 2 (1 = C1 +, 2 = C2 x)
<NRf> = 0 to 500

Example :CURSOR:BAR:POSITION1 1
:CURSOR:BAR:POSITION1?
-> :CURSOR:BAR:POSITION1 1

:CURSor:BAR[:STATe]

Sets or queries the on/off status of the cursor display on the bar graph display.

Syntax :CURSor:BAR[:STATe] {<Boolean>}
:CURSor:BAR:STATe?

Example :CURSOR:BAR:STATE ON
:CURSOR:BAR:STATE?
-> :CURSOR:BAR:STATE 1

:CURSor:BAR:{Y<x>|DY}?

Queries the measured value of the specified cursor on the bar graph display.

Syntax :CURSor:BAR:{Y<x>|DY}?
Y<x> = Y-axis value of the cursor position (Y1 = Y1 +, Y2 +, Y3 +; Y2 = Y1 x, Y2 x, Y3 x)
DY = Difference between the cursor Y-axis values (DY1, DY2, DY3)

Example :CURSOR:BAR:Y1? -> 78.628E+00

Description
- When multiple bar graphs are displayed, the cursor measured values of each bar graph are returned in order.
- If the bar graph cursor display is not turned on, NAN (Not A Number) is returned.

:CURSor:TRENd?

Queries all trend display cursor measurement settings.

Syntax :CURSor:TRENd?

:CURSor:TRENd:LINKage

Sets or queries the on/off status of the cursor position linkage on the trend display.

Syntax :CURSor:TRENd:LINKage {<Boolean>}
:CURSor:TRENd:LINKage?

Example :CURSOR:TREND:LINKAGE OFF
:CURSOR:TREND:LINKAGE?
-> :CURSOR:TREND:LINKAGE 0

:CURSor:TRENd:POSition<x>

Sets or queries the position of the specified cursor on the trend display.

Syntax :CURSor:TRENd:POSition<x> {<NRf>}
:CURSor:TRENd:POSition<x>?
<x> = 1, 2 (1 = C1 +, 2 = C2 x)
<NRf> = 0 to 1601

Example :CURSOR:TREND:POSITION1 160
:CURSOR:TREND:POSITION1?
-> :CURSOR:TREND:POSITION1 160

:CURSor:TRENd[:STATe]

Sets or queries the on/off status of the cursor display on the trend display.

Syntax :CURSor:TRENd[:STATe] {<Boolean>}
:CURSor:TRENd:STATe?

Example :CURSOR:TREND:STATE ON
:CURSOR:TREND:STATE?
-> :CURSOR:TREND:STATE 1

:CURSor:TRENd:TRACe<x>

Sets or queries the target of the specified cursor on the trend display.

Syntax `:CURSor:TRENd:TRACe<x> {<NRf>}`
`:CURSor:TRENd:TRACe<x>?`
<x> = 1, 2 (1 = C1 +, 2 = C2 x)
<NRf> = 1 to 16 (T1 to T16)

Example
```
:CURSOR:TREND:TRACE1 1
:CURSOR:TREND:TRACE1?
-> :CURSOR:TREND:TRACE1 1
```

:CURSor:TRENd:{X<x>|Y<x>|DY}?

Queries the measured value of the specified cursor on the trend display.

Syntax `:CURSor:TRENd:{X<x>|Y<x>|DY}?`
X<x> = Trend time string of the cursor position (X1 = D+, X2 = Dx)
Y<x> = Y-axis value of the cursor position (Y1 = Y+, Y2 = Yx)
DY = Y-axis difference (DY) between the cursors

Example
```
:CURSOR:TREND:X1?
-> "2024/01/01 12:34:56"
:CURSOR:TREND:Y1? -> 78.628E+00
```

Description If the trend cursor display is not turned on, the following results will be returned.
For X<x>: "****/**/** **:**:**" will be returned.
For Y<x> and DY: NAN (Not A Number) will be returned.

:CURSor:WAVE?

Queries all waveform display cursor measurement settings.

Syntax `:CURSor:WAVE?`

:CURSor:WAVE:LINKage

Sets or queries the on/off status of the cursor position linkage on the waveform display.

Syntax `:CURSor:WAVE:LINKage {<Boolean>}`
`:CURSor:WAVE:LINKage?`

Example
```
:CURSOR:WAVE:LINKAGE OFF
:CURSOR:WAVE:LINKAGE?
-> :CURSOR:WAVE:LINKAGE 0
```

:CURSor:WAVE:PATH

Sets or queries the cursor path on the waveform display.

Syntax `:CURSor:WAVE:PATH {MAX|MIN|MID}`
`:CURSor:WAVE:PATH?`

Example
```
:CURSOR:WAVE:PATH MAX
:CURSOR:WAVE:PATH?
-> :CURSOR:WAVE:PATH MAX
```

:CURSor:WAVE:POSition<x>

Sets or queries the position of the specified cursor on the waveform display.

Syntax `:CURSor:WAVE:POSition<x> {<NRf>}`
`:CURSor:WAVE:POSition<x>?`
<x> = 1, 2 (1 = C1 +, 2 = C2 x)
<NRf> = 0 to 800

Example
```
:CURSOR:WAVE:POSITION1 160
:CURSOR:WAVE:POSITION1?
-> :CURSOR:WAVE:POSITION1 160
```

:CURSor:WAVE[:STATe]

Sets or queries the on/off status of the cursor display on the waveform display.

Syntax `:CURSor:WAVE[:STATe] {<Boolean>}`
`:CURSor:WAVE:STATe?`

Example
```
:CURSOR:WAVE:STATE ON
:CURSOR:WAVE:STATE?
-> :CURSOR:WAVE:STATE 1
```

:CURSor:WAVE:TRACe<x>

Sets or queries the target of the specified cursor on the waveform display.

Syntax `:CURSor:WAVE:TRACe<x> {U<x>|I<x>|SPEed|TORQue|AUX<x>}`
`:CURSor:WAVE:TRACe<x>?`
TRACe<x>'s <x> = 1, 2, (1 = C1 +, 2 = C2 x)
U<x> and I<x>'s <x> = 1 to 6 (element)
AUX<x>'s <x> = 1 or 2 (AUX input channel)

Example
```
:CURSOR:WAVE:TRACE1 U1
:CURSOR:WAVE:TRACE1?
-> :CURSOR:WAVE:TRACE1 U1
```

Description
- SPEed and TORQue can only be selected on models with the motor evaluation function (/MTR) option.
- AUX<x> can only be selected on models with the auxiliary input (/AUX) option.

:CURSor:WAVE:{X<x>|DX|PERDt|Y<x>|DY}?

Queries the measured value of the specified cursor on the waveform display.

Syntax `:CURSor:WAVE:{X<x>|DX|PERDt|Y<x>|DY}?`
X<x> = X-axis value of the cursor position (X1 = X+, X2 = Xx)
DX = X-axis difference (DX) between the cursors
PERDt = Value of 1/DT (1/DX) between the cursors
Y<x> = Y-axis value of the cursor position (Y1 = Y+, Y2 = Yx)
DY = Y-axis difference (DY) between the cursors

Example `:CURSOR:WAVE:Y1? -> 78.628E+00`

Description If the waveform cursor display is not turned on, NAN (Not A Number) is returned.

5.6 DISPlay Group

The commands in this group deal with the display.
You can make the same settings and queries that you can make by pressing the keys in the front panel DISPLAY and ITEM & ELEMENT areas.

:DISPlay?

Queries all display settings.

Syntax :DISPlay?

Description Returns all settings that correspond to the current display mode (:DISPlay:MODE).

:DISPlay:BAR?

Queries all bar graph display settings.

Syntax :DISPlay:BAR?

Description The bar graph display is only available on models with the harmonic measurement (/G5 or /G6) option.

:DISPlay:BAR:FORMat

Sets or queries the bar graph display format.

Syntax :DISPlay:BAR:FORMat {SINGle|DUAL|TRIad}
:DISPlay:BAR:FORMat?

Example :DISPLAY:BAR:FORMAT SINGLE
:DISPLAY:BAR:FORMAT?
-> :DISPLAY:BAR:FORMAT SINGLE

:DISPlay:BAR:ITEM<x>?

Queries all the display settings of the specified bar graph.

Syntax :DISPlay:BAR:ITEM<x>?
<x> = 1 to 3 (item number)

:DISPlay:BAR:ITEM<x>[:FUNCtion]

Sets or queries the function and element of the specified bar graph item.

Syntax :DISPlay:BAR:ITEM<x>[:FUNCtion] {<Function>,<Element>}
:DISPlay:BAR:ITEM<x>:FUNCtion?
<x> = 1 to 3 (item number)
<Function> = {U|I|P|S|Q|LAMBda|PHI|PHIU|PHII|Z|RS|XS|RP|XP}
<Element> = 1 to 6

Example :DISPLAY:BAR:ITEM1 U,1
:DISPLAY:BAR:ITEM1?
-> :DISPLAY:BAR:ITEM1 U,1

Description For information about the options available for <Function>, see “Numeric list data functions” in “Function option list” on page 5-47.

:DISPlay:BAR:ITEM<x>:SCALing?

Queries all scaling settings for the specified bar graph.

Syntax :DISPlay:BAR:ITEM<x>:SCALing?
<x> = 1 to 3 (item number)

:DISPlay:BAR:ITEM<x>:SCALing:MODE

Sets or queries the scaling mode of the specified bar graph.

Syntax :DISPlay:BAR:ITEM<x>:SCALing:MODE {FIXed|MANual}
:DISPlay:BAR:ITEM<x>:SCALing:MODE?
<x> = 1 to 3 (item number)

Example :DISPLAY:BAR:ITEM1:SCALING:MODE FIXED
:DISPLAY:BAR:ITEM1:SCALING:MODE?
-> :DISPLAY:BAR:ITEM1:SCALING:MODE FIXED

:DISPlay:BAR:ITEM<x>:SCALing:VALue

Sets or queries the upper limit of the manual scaling of the specified bar graph.

Syntax :DISPlay:BAR:ITEM<x>:SCALing:VALue {<NRf>}
:DISPlay:BAR:ITEM<x>:SCALing:VALue?
<x> = 1 to 3 (item number)
<NRf> = 0 to 9.999E+12

Example :DISPLAY:BAR:ITEM1:SCALING:VALUE 100
:DISPLAY:BAR:ITEM1:SCALING:VALUE?
-> :DISPLAY:BAR:ITEM1:SCALING:VALUE 100.0E+00

Description
- This command is valid when the scaling mode of the bar graph (:DISPlay:BAR:ITEM<x>:SCALing:MODE) is set to MANual.
- This command only sets the upper limit. The lower limit is determined automatically (as shown below) according to the vertical scaling mode (:DISPlay:BAR:ITEM<x>:SCALing:VERTical).
  - When the mode is LINear: 0 when the X-axis position (:DISPlay:BAR:ITEM<x>:SCALing:XAXis) is “BOTTom;” the negative value of the upper limit when the X-axis position is “CENTer”
  - When the mode is LOG: The upper limit/10000

:DISPlay:BAR:ITEM<x>:SCALing:VERTical

Sets or queries the vertical scaling mode of the specified bar graph.

Syntax
```
:DISPlay:BAR:ITEM<x>:SCALing:
VERTical {LINear|LOG}
:DISPlay:BAR:ITEM<x>:SCALing:
VERTical?
```
<x> = 1 to 3 (item number)

Example
```
:DISPLAY:BAR:ITEM1:SCALING:
VERTICAL LOG
:DISPLAY:BAR:ITEM1:SCALING:
VERTICAL?
-> :DISPLAY:BAR:ITEM1:SCALING:
VERTICAL LOG
```

Description This command is valid when the scaling mode of the bar graph (:DISPlay:BAR:ITEM<x>:SCALing:MODE) is set to MANual.

:DISPlay:BAR:ITEM<x>:SCALing:XAXis

Sets or queries the position of the X axis of the specified bar graph.

Syntax
```
:DISPlay:BAR:ITEM<x>:SCALing:
XAXis {BOTTom|CENTer}
:DISPlay:BAR:ITEM<x>:SCALing:
XAXis?
```
<x> = 1 to 3 (item number)

Example
```
:DISPLAY:BAR:ITEM1:SCALING:
XAXIS BOTTOM
:DISPLAY:BAR:ITEM1:SCALING:XAXIS?
-> :DISPLAY:BAR:ITEM1:SCALING:
XAXIS BOTTOM
```

Description This command is valid when the scaling mode of the bar graph (:DISPlay:BAR:ITEM<X>:SCALing:MODE) is set to MANual and the vertical scaling mode of the bar graph (:DISPlay:BAR:ITEM<x>:SCALing:VERTical) is set to LINear.

:DISPlay:BAR:ORDer

Sets or queries the displayed starting and ending harmonic orders of the bar graphs.

Syntax
```
:DISPlay:BAR:ORDer {<NRf>,<NRf>}
:DISPlay:BAR:ORDer?
```
First <NRf> = 0 to 490 (displayed starting harmonic order)

Second <NRf> = 10 to 500 (displayed ending harmonic order)

Example
```
:DISPLAY:BAR:ORDER 1,100
:DISPLAY:BAR:ORDER?
-> :DISPLAY:BAR:ORDER 1,100
```

Description
- Set the starting harmonic order and then the ending harmonic order.
- Set the ending harmonic order to a value greater than or equal to that of the starting harmonic order + 10.

:DISPlay:HSPeed?

Queries all high speed data capturing display settings.

Syntax
```
:DISPlay:HSPeed?
```

:DISPlay:HSPeed:COLumn?

Queries all column settings of the high speed data capturing mode.

Syntax
```
:DISPlay:HSPeed:COLumn?
```

:DISPlay:HSPeed:COLumn:ITEM<x>

Sets or queries the specified column display item of the high speed data capturing mode.

Syntax
```
:DISPlay:HSPeed:COLumn:
ITEM<x> {NONE|<Element>}
:DISPlay:HSPeed:COLumn:ITEM<x>?
```
<x> = 1 to 6 (column number)

<Element> = {<NRf>|SIGMa|SIGMB|SIGMC}
(<NRf> = 1 to 6)

Example
```
:DISPLAY:HSPEED:COLUMN:ITEM1 1
:DISPLAY:HSPEED:COLUMN:ITEM1?
-> :DISPLAY:HSPEED:COLUMN:ITEM1 1
```

:DISPlay:HSPeed:COLumn:NUMber

Sets or queries the number of columns of the high speed data capturing mode.

Syntax
```
:DISPlay:HSPeed:COLumn:
NUMber {<NRf>}
:DISPlay:HSPeed:COLumn:NUMber?
```
<NRf> = 4, 6

Example
```
:DISPLAY:HSPEED:COLUMN:NUMBER 4
:DISPLAY:HSPEED:COLUMN:NUMBER?
-> :DISPLAY:HSPEED:COLUMN:NUMBER 4
```

:DISPlay:HSPeed:COLumn:RESet

Resets the column display items to their default values on the high speed data capturing mode.

Syntax
```
:DISPlay:HSPeed:COLumn:RESet
```

Example
```
:DISPLAY:HSPEED:COLUMN:RESET
```

:DISPlay:HSPeed:FRAMe

Sets or queries the on/off status of the high speed data capturing mode's data section frame.

Syntax
```
:DISPlay:HSPeed:FRAMe {<Boolean>}
:DISPlay:HSPeed:FRAMe?
```

Example
```
:DISPLAY:HSPEED:FRAME ON
:DISPLAY:HSPEED:FRAME?
-> :DISPLAY:HSPEED:FRAME 1
```

Description This command performs the same setting as the ":DISPlay:NUMeric:FRAMe" command.

:DISPlay:HSPeed:PAGE

Sets or queries the displayed page of the high speed data capturing mode.

Syntax `:DISPlay:HSPeed:PAGE {<NRf>}`
`:DISPlay:HSPeed:PAGE?`
<NRf> = 1 to 2 (page number)
<NRf> = 1 to 4 (on models with the motor evaluation function (/MTR) option or the auxiliary input (/AUX) option.)

Example
```
:DISPLAY:HSPEED:PAGE 1
:DISPLAY:HSPEED:PAGE?
-> :DISPLAY:HSPEED:PAGE 1
```

:DISPlay:HSPeed:POVer

Sets or queries the on/off status of the display of peak over-range information in high speed data capturing mode.

Syntax `:DISPlay:HSPeed:POVer {<Boolean>}`
`:DISPlay:HSPeed:POVer?`

Example
```
:DISPLAY:HSPEED:POVER OFF
:DISPLAY:HSPEED:POVER?
-> :DISPLAY:HSPEED:POVER 0
```

:DISPlay:INFOrmation?

Queries all setup parameter list display settings.

Syntax `:DISPlay:INFOrmation?`

:DISPlay:INFOrmation:PAGE

Sets or queries the displayed page of the setup parameter list display.

Syntax `:DISPlay:INFOrmation:PAGE {POWer|RANGe|<NRf>}`
`:DISPlay:INFOrmation:PAGE?`
POWer (or <NRf> = 1) = Tables of each element's measurement conditions (Power Element Settings)
RANGe (or <NRf> = 2) = Indicators of each element's voltage and current range settings (Range Settings)

Example
```
:DISPLAY:INFORMATION:PAGE POWER
:DISPLAY:INFORMATION:PAGE?
-> :DISPLAY:INFORMATION:PAGE POWER
```

:DISPlay:INFOrmation[:STATe]

Sets or queries the on/off status of the setup parameter list display.

Syntax `:DISPlay:INFOrmation[:STATe] {<Boolean>}`
`:DISPlay:INFOrmation:STATe?`

Example
```
:DISPLAY:INFORMATION:STATE ON
:DISPLAY:INFORMATION:STATE?
-> :DISPLAY:INFORMATION:STATE 1
```

:DISPlay:MODE

Sets or queries the display mode.

Syntax `:DISPlay:MODE {NUMeric|WAVE|TRENd|BAR|VECTor|NWAVe|NTRend|NBAR|NVECtor|WNUMeric|WTRend|WBAR|WVECtor|TNUMeric|TWAVe|TBAR|TVECtor|HSPeed}`
`:DISPlay:MODE?`
NUMeric = Numeric display
WAVE = Waveform display
TRENd = Trend display
BAR = Bar graph display
VECTor = Vector display
NWAVe = Numeric and waveform displays
NTRend = Numeric and trend displays
NBAR = Numeric and bar graph displays
NVECtor = Numeric and vector displays
WNUMeric = Waveform and numeric displays
WTRend = Waveform and trend displays
WBAR = Waveform and bar graph displays
WVECtor = Waveform and vector displays
TNUMeric = Trend and numeric displays
TWAVe = Trend and waveform displays
TBAR = Trend and bar graph displays
TVECtor = Trend and vector displays
HSPeed = High speed data capturing mode display (numeric display)

Example
```
:DISPLAY:MODE NUMERIC
:DISPLAY:MODE?
-> :DISPLAY:MODE NUMERIC
```

Description BAR, VECTor, NBAR, NVECtor, WBAR, WVECtor, TBAR, and TVECtor can only be selected on models with the harmonic measurement (/G5 or /G6) option.

:DISPlay:NUMeric?

Queries all numeric display settings.

Syntax `:DISPlay:NUMeric?`

:DISPlay:NUMeric:CUSTom?

Queries all numeric display settings in custom display mode.

Syntax `:DISPlay:NUMeric:CUSTom?`

:DISPlay:NUMeric:CUSTom:FILE:CDIRectory

Changes the directory that files are loaded from or saved to for the numeric display in custom display mode.

Syntax `:DISPlay:NUMeric:CUSTom:FILE:CDIRectory {<String>}`
<String> = Directory name

Example
```
:DISPLAY:NUMERIC:CUSTOM:FILE:CDIRECTORY "CUSTOM"
```

Description Specify ".." to move up to the parent directory.

:DISPlay:NUMeric:CUSTom:FILE:DRIVe

Sets the drive that files are loaded from or saved to for the numeric display in custom display mode.

Syntax `:DISPlay:NUMeric:CUSTom:FILE:DRIVe {USER|RAM|USB[,<NRf>]|NETWork}`
USER = Internal memory drive
RAM = Internal RAM disk
USB = USB memory device drive,
<NRf> = 0 or 1 (drive number)
NETWork = Network drive

Example `:DISPLAY:NUMERIC:CUSTOM:FILE:DRIVE USB,0`

:DISPlay:NUMeric:CUSTom:FILE:FREE?

Queries the amount of free space (in bytes) on the drive that files are loaded from or saved to for the numeric display in custom display mode.

Syntax `:DISPlay:NUMeric:CUSTom:FILE:FREE?`

Example `:DISPLAY:NUMERIC:CUSTOM:FILE:FREE?`
`-> 20912128`

:DISPlay:NUMeric:CUSTom:FILE:LOAD:ABORt

Aborts a file loading operation for the numeric display in custom display mode.

Syntax `:DISPlay:NUMeric:CUSTom:FILE:LOAD:ABORt`

Example `:DISPLAY:NUMERIC:CUSTOM:FILE:LOAD:ABORT`

:DISPlay:NUMeric:CUSTom:FILE:LOAD:BMP

Loads the specified background file for the numeric display in custom display mode.

Syntax `:DISPlay:NUMeric:CUSTom:FILE:LOAD:BMP {<String>}`
<String> = File name

Example `:DISPLAY:NUMERIC:CUSTOM:FILE:LOAD:BMP "CUSTOM1"`

Description
- Specify the file name without its extension (.bmp).
- This command is an overlap command.

:DISPlay:NUMeric:CUSTom:FILE:LOAD:BOTH

Loads the specified display configuration and background files for the numeric display in custom display mode.

Syntax `:DISPlay:NUMeric:CUSTom:FILE:LOAD:BOTH {<String>}`
<String> = File name

Example `:DISPLAY:NUMERIC:CUSTOM:FILE:LOAD:BOTH "CUSTOM1"`

Description
- Specify the file name without an extension.
- This command is an overlap command.

:DISPlay:NUMeric:CUSTom:FILE:LOAD:ITEM

Loads the specified display configuration file for the numeric display in custom display mode.

Syntax `:DISPlay:NUMeric:CUSTom:FILE:LOAD:ITEM {<String>}`
<String> = File name

Example `:DISPLAY:NUMERIC:CUSTOM:FILE:LOAD:ITEM "CUSTOM1"`

Description
- Specify the file name without its extension (.txt).
- This command is an overlap command.

:DISPlay:NUMeric:CUSTom:FILE:PATH?

Queries the absolute path of the directory that files are loaded from or saved to for the numeric display in custom display mode.

Syntax `:DISPlay:NUMeric:CUSTom:FILE:PATH?`

Example `:DISPLAY:NUMERIC:CUSTOM:FILE:PATH?`
`-> "USB-0/CUSTOM"`

:DISPlay:NUMeric:CUSTom:FILE:SAVE:ANAMing

Sets or queries the automatic file name generation feature for saving display configuration files of the numeric display in custom display mode.

Syntax `:DISPlay:NUMeric:CUSTom:FILE:SAVE:ANAMing {OFF|NUMBering|DATE}`
`:DISPlay:NUMeric:CUSTom:FILE:SAVE:ANAMing?`

Example `:DISPLAY:NUMERIC:CUSTOM:FILE:SAVE:ANAMING NUMBERING`
`:DISPLAY:NUMERIC:CUSTOM:FILE:SAVE:ANAMING?`
`-> :DISPLAY:NUMERIC:CUSTOM:FILE:SAVE:ANAMING NUMBERING`

:DISPlay:NUMeric:CUSTom:FILE:SAVE:ITEM

Saves the specified display configuration file for the numeric display in custom display mode.

Syntax `:DISPlay:NUMeric:CUSTom:FILE:SAVE:ITEM {<String>}`
<String> = File name

Example `:DISPLAY:NUMERIC:CUSTOM:FILE:SAVE:ITEM "CUSTOM1"`

Description
- Specify the file name without its extension (.txt).
- This command is an overlap command.

:DISPlay:NUMeric:CUSTom:ITEM<x>?

Queries all the settings of the specified display item of the numeric display in custom display mode.

Syntax `:DISPlay:NUMeric:CUSTom:ITEM<x>?`
<x> = 1 to 192 (item number)

:DISPlay:NUMeric:CUSTom:ITEM<x>:COLor

Sets or queries the font color of the specified display item of the numeric display in custom display mode.

Syntax
```
:DISPlay:NUMeric:CUSTom:ITEM<x>:
COLor {YELLow|GREen|MAGenta|CYAN|
RED|ORANge|LBLue|PURPle|BLUE|PINK|
LGReen|DBLue|BGReen|SPINk|MGReen|
GRAY|WHITE|DGRAY|BGRAY|BLACk}
:DISPlay:NUMeric:CUSTom:ITEM<x>:
COLor?
```
<x> = 1 to 192 (item number)
YELLow = Yellow
GREen = Green
MAGenta = Magenta
CYAN = Cyan
RED = Red
ORANge = Orange
LBLue = Light blue
PURPle = Purple
BLUE = Blue
PINK = Pink
LGReen = Light green
DBLue = Dark blue
BGReen = Blue green
SPINk = Salmon pink
MGReen = Mild green
GRAY = Gray
WHITE = White
DGRAY = Dark gray
BGRAY = Blue gray
BLACk = Black

Example
```
:DISPLAY:NUMERIC:CUSTOM:ITEM1:
COLOR WHITE
:DISPLAY:NUMERIC:CUSTOM:ITEM1:COLOR?
-> :DISPLAY:NUMERIC:CUSTOM:ITEM1:
COLOR WHITE
```

:DISPlay:NUMeric:CUSTom:ITEM<x>[:FUNCtion]

Sets or queries the display item (numeric item or string) of the numeric display in custom display mode.

Syntax
```
:DISPlay:NUMeric:CUSTom:
ITEM<x>[:FUNCtion] {<Function>
[,<Element>][,<Order>]|<String>}
:DISPlay:NUMeric:CUSTom:
ITEM<x>[:FUNCtion]?
```
<x> = 1 to 192 (item number)

When setting a numeric item

<Function> = {URMS|IRMS|P|S|Q|...}
<Element> = {<NRf>|SIGMa|SIGMB|SIGMC}
(<NRf> = 1 to 6)
<Order> = {TOTal|DC|<NRf>}
(<NRf> = 1 to 500)

When setting a string

<String> = Up to 16 characters

Example

When setting a numeric item
```
:DISPLAY:NUMERIC:CUSTOM:ITEM1:
FUNCTION URMS,1
:DISPLAY:NUMERIC:CUSTOM:ITEM1:
FUNCTION?
-> :DISPLAY:NUMERIC:CUSTOM:ITEM1:
FUNCTION URMS,1
:DISPLAY:NUMERIC:CUSTOM:ITEM1:
FUNCTION UK,1,1
:DISPLAY:NUMERIC:CUSTOM:ITEM1:
FUNCTION?
-> :DISPLAY:NUMERIC:CUSTOM:ITEM1:
FUNCTION UK,1,1
```
When setting a string
```
:DISPLAY:NUMERIC:CUSTOM:ITEM1:
FUNCTION "YOKOGAWA"
:DISPLAY:NUMERIC:CUSTOM:ITEM1:
FUNCTION?
-> :DISPLAY:NUMERIC:CUSTOM:ITEM1:
FUNCTION "YOKOGAWA"
```

Description Set a numeric item or a string as a display item.

When setting a numeric item

- For information about the options available for <Function>, see “Numeric data functions” in “Function option list” on page 5-44.
- If <Element> is omitted, the element is set to 1.
- If <Order> is omitted, the order is set to TOTal.
- <Element> and <Order> are omitted from responses to functions that do not need them.

When setting a string

You can display any string that you want, for example, the header or unit of a numeric item.

:DISPlay:NUMeric:CUSTom:ITEM<x>:POSition

Sets or queries the display position of the specified display item of the numeric display in custom display mode.

Syntax
```
:DISPlay:NUMeric:CUSTom:ITEM<x>:
POSition {<NRf>,<NRf>}
:DISPlay:NUMeric:CUSTom:ITEM<x>:
POSition?
```
<x> = 1 to 192 (item number)
First <NRf> = 0 to 800 (X coordinate)
Second <NRf> = 0 to 672 (Y coordinate)

Example
```
:DISPLAY:NUMERIC:CUSTOM:ITEM1:
POSITION 0,0
:DISPLAY:NUMERIC:CUSTOM:ITEM1:
POSITION?
-> :DISPLAY:NUMERIC:CUSTOM:ITEM1:
POSITION 0,0
```

Description The upper-left corner of the numeric data display area is the origin, and the specified coordinate refers to the upper left of the display item.

:DISPlay:NUMeric:CUSTom:ITEM<x>:SIZE

Sets or queries the font size of the specified display item of the numeric display in custom display mode.

Syntax
```
:DISPlay:NUMeric:CUSTom:ITEM<x>:
SIZE {<NRf>}
:DISPlay:NUMeric:CUSTom:ITEM<x>:SIZE?
```
<x> = 1 to 192 (item number)
<NRf> = 14, 16, 20, 24, 32, 48, 64, 96, 128

Example
```
:DISPLAY:NUMERIC:CUSTOM:ITEM1:
SIZE 20
:DISPLAY:NUMERIC:CUSTOM:ITEM1:
SIZE?
-> :DISPLAY:NUMERIC:CUSTOM:ITEM1:
SIZE 20
```

:DISPlay:NUMeric:CUSTom:PAGE

Sets or queries the displayed page of the numeric display in custom display mode.

Syntax
```
:DISPlay:NUMeric:CUSTom:PAGE {<NRf>}
:DISPlay:NUMeric:CUSTom:PAGE?
```
<NRf> = 1 to 12 (page number)

Example
```
:DISPLAY:NUMERIC:CUSTOM:PAGE 1
:DISPLAY:NUMERIC:CUSTOM:PAGE?
-> :DISPLAY:NUMERIC:CUSTOM:PAGE 1
```

Description The maximum page number that can be displayed is determined by the total number of display items and the number of items per page.

:DISPlay:NUMeric:CUSTom:PERPage

Sets or queries the number of items displayed per page of the numeric display in custom display mode.

Syntax
```
:DISPlay:NUMeric:CUSTom:
PERPage {<NRf>}
:DISPlay:NUMeric:CUSTom:
PERPage?
```
<NRf> = 1 to the total number of display items

Example
```
:DISPLAY:NUMERIC:CUSTOM:PERPAGE 5
:DISPLAY:NUMERIC:CUSTOM:PERPAGE?
-> :DISPLAY:NUMERIC:CUSTOM:PERPAGE 5
```

Description The minimum number of items that can be displayed per page is “total number of display items (:DISPlay:NUMeric:CUSTom:TOTal)/12.”

:DISPlay:NUMeric:CUSTom:TOTal

Sets or queries the total number of display items of the numeric display in custom display mode.

Syntax
```
:DISPlay:NUMeric:CUSTom:
TOTal {<NRf>}
:DISPlay:NUMeric:CUSTom:TOTal?
```
<NRf> = 1 to 192 (number of items)

Example
```
:DISPLAY:NUMERIC:CUSTOM:TOTAL 20
:DISPLAY:NUMERIC:CUSTOM:TOTAL?
-> :DISPLAY:NUMERIC:CUSTOM:TOTAL 20
```

Description The maximum number of total display items is “number of display items per page (:DISPlay:NUMeric:CUSTom:PERPage) × 12.”

:DISPlay:NUMeric:FRAMe

Sets or queries the on/off status of the numeric display’s data section frame.

Syntax
```
:DISPlay:NUMeric:FRAMe {<Boolean>}
:DISPlay:NUMeric:FRAMe?
```

Example
```
:DISPLAY:NUMERIC:FRAME ON
:DISPLAY:NUMERIC:FRAME?
-> :DISPLAY:NUMERIC:FRAME 1
```

:DISPlay:NUMeric:NORMal?

Queries all numeric display settings.

Syntax
```
:DISPlay:NUMeric:NORMal?
```

Description Returns all settings that correspond to the current numeric display mode (:DISPlay:NUMeric[:NORMal]:FORMat).

:DISPlay:NUMeric[:NORMal]:ALL?

Queries all settings of the numeric display in All Items display mode.

Syntax
```
:DISPlay:NUMeric[:NORMal]:ALL?
```

:DISPlay:NUMeric[:NORMal]:ALL:COLumn?

Queries all column settings of the numeric display in All Items display mode.

Syntax :DISPlay:NUMeric[:NORMal]:ALL:COLumn?

Description Column display settings are only valid on models that have five or more elements.

:DISPlay:NUMeric[:NORMal]:ALL:COLumn:DAELem

Sets or queries the on/off status of the column display all feature of the numeric display in All Items display mode.

Syntax :DISPlay:NUMeric[:NORMal]:ALL:COLumn:DAELem {<Boolean>}
:DISPlay:NUMeric[:NORMal]:ALL:COLumn:DAELem?

Example :DISPLAY:NUMERIC:NORMAL:ALL:COLUMN:DAELEM ON
:DISPLAY:NUMERIC:NORMAL:ALL:COLUMN:DAELEM?
-> :DISPLAY:NUMERIC:NORMAL:ALL:COLUMN:DAELEM 1

Description This feature (Display All Elements) automatically decreases the font size to display all columns when the number of columns that should be displayed exceeds 6 according to the wiring system setting (element/Σ).

:DISPlay:NUMeric[:NORMal]:ALL:COLumn:SCRoll

Sets or queries the on/off status of column scrolling of the numeric display in All Items display mode.

Syntax :DISPlay:NUMeric[:NORMal]:ALL:COLumn:SCRoll {<NRf>}
:DISPlay:NUMeric[:NORMal]:ALL:COLumn:SCRoll?
<NRf> = 0 to 3 (scroll amount)

Example :DISPLAY:NUMERIC:NORMAL:ALL:COLUMN:SCROLL 0
:DISPLAY:NUMERIC:NORMAL:ALL:COLUMN:SCROLL?
-> :DISPLAY:NUMERIC:NORMAL:ALL:COLUMN:SCROLL 0

Description This command is valid when the column display all feature (:DISPlay:NUMeric[:NORMal]:ALL:COLumn:DAELem) is set to OFF.

:DISPlay:NUMeric[:NORMal]:ALL:CURSor

Sets or queries the cursor position on the numeric display in All Items display mode.

Syntax :DISPlay:NUMeric[:NORMal]:ALL:CURSor {<Function>}
:DISPlay:NUMeric[:NORMal]:ALL:CURSor?
<Function> = {URMS|IRMS|P|S|Q|...}

Example :DISPLAY:NUMERIC:NORMAL:ALL:CURSOR P
:DISPLAY:NUMERIC:NORMAL:ALL:CURSOR?
-> :DISPLAY:NUMERIC:NORMAL:ALL:CURSOR P

Description
- Use the function name to specify the cursor position.
- For information about the options available for <Function>, see “Numeric data functions” in “Function option list” on page 5-44.

:DISPlay:NUMeric[:NORMal]:ALL:ORDer

Sets or queries the displayed harmonic order on the harmonic measurement function display page of the numeric display in All Items display mode.

Syntax :DISPlay:NUMeric[:NORMal]:ALL:ORDer {<Order>}
:DISPlay:NUMeric[:NORMal]:ALL:ORDer?
<Order> = {TOTal|DC|<NRf>}
(<NRf> = 1 to 500)

Example :DISPLAY:NUMERIC:NORMAL:ALL:ORDER 1
:DISPLAY:NUMERIC:NORMAL:ALL:ORDER?
-> :DISPLAY:NUMERIC:NORMAL:ALL:ORDER 1

Description
- This is only valid on models with the harmonic measurement (/G5 or /G6) option.
- This command is valid when the displayed page number (:DISPlay:NUMeric[:NORMal]:ALL:PAGE) of the numeric display in All Items display mode is 9 or 10.

:DISPlay:NUMeric[:NORMal]:ALL:PAGE

Sets or queries the displayed page of the numeric display in All Items display mode.

Syntax :DISPlay:NUMeric[:NORMal]:ALL:PAGE {<NRf>}
:DISPlay:NUMeric[:NORMal]:ALL:PAGE?
<NRf> = 1 to 8 (page number)
<NRf> = 1 to 12 (when the harmonic measurement [/G5 or /G6] option is installed)

Example :DISPLAY:NUMERIC:NORMAL:ALL:PAGE 1
:DISPLAY:NUMERIC:NORMAL:ALL:PAGE?
-> :DISPLAY:NUMERIC:NORMAL:ALL:PAGE 1

Description When the page number is set, the cursor position moves to the beginning of the page.

:DISPlay:NUMeric[:NORMal]:FORMat

Sets or queries the numeric display format.

Syntax
```
:DISPlay:NUMeric[:NORMal]:
FORMat {VAL4|VAL8|VAL16|MATRix|ALL|
SINGle|DUAL|CUSTom}
:DISPlay:NUMeric[:NORMal]:FORMat?
```
Example
```
:DISPLAY:NUMERIC:NORMAL:FORMAT VAL4
:DISPLAY:NUMERIC:NORMAL:FORMAT?
-> :DISPLAY:NUMERIC:NORMAL:
FORMAT VAL4
```
Description
- The numeric data is displayed in the following ways for each format:
  {VAL4|VAL8|VAL16} = Numeric display items are displayed in order by their item numbers. (The numbers in these options indicate the number of items that are displayed on a single screen/page.)
  MATRix = Selected functions are displayed in order by element.
  ALL = All functions are displayed in order by element.
  SINGle = One list display item is listed by separating the data into even and odd harmonic orders.
  DUAL = Two list display items are listed in order by harmonic order.
  CUSTom = The specified numeric display items are displayed on the specified bitmap background.
- SINGle and DUAL can only be selected on models with the harmonic measurement (/G5 or /G6) option.

:DISPlay:NUMeric[:NORMal]:LIST?

Queries all numeric display settings in the list display modes.

Syntax
```
:DISPlay:NUMeric[:NORMal]:LIST?
```
Description The list display is only available on models with the harmonic measurement (/G5 or /G6) option.

:DISPlay:NUMeric[:NORMal]:LIST:CURSor

Sets or queries the cursor position on the numeric display in the list display modes.

Syntax
```
:DISPlay:NUMeric[:NORMal]:LIST:
CURSor {HEADer|ORDer}
:DISPlay:NUMeric[:NORMal]:LIST:CURSor?
```
HEADer = The cursor moves to the header section (data concerning all the harmonics; left side of the screen).
ORDer = The cursor moves to the data section (numeric data of each harmonic; right side of the screen).

Example
```
:DISPLAY:NUMERIC:NORMAL:LIST:
CURSOR ORDER
:DISPLAY:NUMERIC:NORMAL:LIST:
CURSOR?
-> :DISPLAY:NUMERIC:NORMAL:LIST:
CURSOR ORDER
```

:DISPlay:NUMeric[:NORMal]:LIST:HEADer

Sets or queries the cursor position of the header section on the numeric display in the list display modes.

Syntax
```
:DISPlay:NUMeric[:NORMal]:LIST:
HEADer {<NRf>}
:DISPlay:NUMeric[:NORMal]:LIST:HEADer?
```
<NRf> = 1 to 155 (header row)

Example
```
:DISPLAY:NUMERIC:NORMAL:LIST:
HEADER 1
:DISPLAY:NUMERIC:NORMAL:LIST:
HEADER?
-> :DISPLAY:NUMERIC:NORMAL:LIST:
HEADER 1
```
Description This command is valid when the cursor position (:DISPlay:NUMeric[:NORMal]:LIST:CURSor) on the numeric display in the list display modes is set to HEADer.

:DISPlay:NUMeric[:NORMal]:LIST:ITEM<x>

Sets or queries the specified display item (function and element) on the numeric display in the list display modes.

Syntax
```
:DISPlay:NUMeric[:NORMal]:LIST:
ITEM<x> {<Function>,<Element>}
:DISPlay:NUMeric[:NORMal]:LIST:
ITEM<x>?
```
<x> = 1 or 2 (item number)
<Function> = {U|I|P|S|Q|LAMBda|PHI|PHIU|PHII|Z|RS|XS|RP|XP}
<Element> = {<NRf>|SIGMa|SIGMB|SIGMC} (<NRf> = 1 to 6)

Example
```
:DISPLAY:NUMERIC:NORMAL:LIST:
ITEM1 U,1
:DISPLAY:NUMERIC:NORMAL:LIST:
ITEM1?
-> :DISPLAY:NUMERIC:NORMAL:LIST:
ITEM1 U,1
```
Description For information about the options available for <Function>, see “Numeric list data functions” in “Function option list” on page 5-47.

:DISPlay:NUMeric[:NORMal]:LIST:ORDer

Sets or queries the harmonic order cursor position of the data section on the numeric display in the list display modes.

Syntax
```
:DISPlay:NUMeric[:NORMal]:LIST:ORDer {<NRf>}
:DISPlay:NUMeric[:NORMal]:LIST:ORDer?
```
<NRf> = 1 to 500 (harmonic order)

Example
```
:DISPLAY:NUMERIC:NORMAL:LIST:ORDER 1
:DISPLAY:NUMERIC:NORMAL:LIST:ORDER?
-> :DISPLAY:NUMERIC:NORMAL:LIST:ORDER 1
```

Description This command is valid when the cursor position (:DISPlay:NUMeric[:NORMal]:LIST:CURSor) on the numeric display in the list display modes is set to ORDer.

:DISPlay:NUMeric[:NORMal]:MATRix?

Queries all numeric display settings in matrix display mode.

Syntax
```
:DISPlay:NUMeric[:NORMal]:MATRix?
```

:DISPlay:NUMeric[:NORMal]:MATRix:COLumn?

Queries all column settings of the numeric display in matrix display mode.

Syntax
```
:DISPlay:NUMeric[:NORMal]:MATRix:COLumn?
```

:DISPlay:NUMeric[:NORMal]:MATRix:COLumn:ITEM<x>

Sets or queries the specified column display item of the numeric display in matrix display mode.

Syntax
```
:DISPlay:NUMeric:[:NORMal]:MATRix:COLumn:ITEM<x> {NONE|<Element>}
:DISPlay:NUMeric:[:NORMal]:MATRix:COLumn:ITEM<x>?
```
<x> = 1 to 6 (column number)
<Element> = {<NRf>|SIGMa|SIGMB|SIGMC} (<NRf> = 1 to 6)

Example
```
:DISPLAY:NUMERIC:NORMAL:MATRIX:COLUMN:ITEM1 1
:DISPLAY:NUMERIC:NORMAL:MATRIX:COLUMN:ITEM1?
-> :DISPLAY:NUMERIC:NORMAL:MATRIX:COLUMN:ITEM1 1
```

:DISPlay:NUMeric[:NORMal]:MATRix:COLumn:NUMber

Sets or queries the number of columns of the numeric display in matrix display mode.

Syntax
```
:DISPlay:NUMeric[:NORMal]:MATRix:COLumn:NUMber {<NRf>}
:DISPlay:NUMeric[:NORMal]:MATRix:COLumn:NUMber?
```
<NRf> = 4, 6

Example
```
:DISPLAY:NUMERIC:NORMAL:MATRIX:COLUMN:NUMBER 4
:DISPLAY:NUMERIC:NORMAL:MATRIX:COLUMN:NUMBER?
-> :DISPLAY:NUMERIC:NORMAL:MATRIX:COLUMN:NUMBER 4
```

:DISPlay:NUMeric[:NORMal]:MATRix:COLumn:RESet

Resets the column display items to their default values on the numeric display in matrix display mode.

Syntax
```
:DISPlay:NUMeric[:NORMal]:MATRix:COLumn:RESet
```

Example
```
:DISPLAY:NUMERIC:NORMAL:MATRIX:COLUMN:RESET
```

:DISPlay:NUMeric[:NORMal]:MATRix:CURSor

Sets or queries the cursor position on the numeric display in matrix display mode.

Syntax
```
:DISPlay:NUMeric[:NORMal]:MATRix:CURSor {<NRf>}
:DISPlay:NUMeric[:NORMal]:MATRix:CURSor?
```
<NRf> = 1 to 81 (item number)

Example
```
:DISPLAY:NUMERIC:NORMAL:MATRIX:CURSOR 1
:DISPLAY:NUMERIC:NORMAL:MATRIX:CURSOR?
-> :DISPLAY:NUMERIC:NORMAL:MATRIX:CURSOR 1
```

Description Use an item number to specify the cursor position.

:DISPlay:NUMeric[:NORMal]:MATRix:ITEM<x>

Sets or queries the specified display item (function and harmonic order) on the numeric display in matrix display mode.

Syntax
```
:DISPlay:NUMeric[:NORMal]:MATRix:
ITEM<x> {NONE|<Function>[,<Element>]
[,<Order>]}
:DISPlay:NUMeric[:NORMal]:MATRix:
ITEM<x>?
```
<x> = 1 to 81 (item number)
NONE = No display item
<Function> = {URMS|IRMS|P|S|Q|...}
<Element> = {<NRf>|SIGMa|SIGMB|SIGMC}
(<NRf> = 1 to 6)
<Order> = {TOTal|DC|<NRf>}
(<NRf> = 1 to 500)

Example
```
:DISPLAY:NUMERIC:NORMAL:MATRIX:
ITEM1 URMS
:DISPLAY:NUMERIC:NORMAL:MATRIX:ITEM1?
-> :DISPLAY:NUMERIC:NORMAL:MATRIX:
ITEM1 URMS,1
:DISPLAY:NUMERIC:NORMAL:MATRIX:
ITEM1 UK,1,1
:DISPLAY:NUMERIC:NORMAL:MATRIX:ITEM1?
-> :DISPLAY:NUMERIC:NORMAL:MATRIX:
ITEM1 UK,1,1
```

Description
- For information about the options available for <Function>, see “Numeric data functions” in “Function option list” on page 5-44.
- The <Element> setting has no effect on the display. If <Element> is omitted, the element is set to 1.
- If <Order> is omitted, the order is set to TOTal.
- <Element> and <Order> are omitted from responses to functions that do not need them.

:DISPlay:NUMeric[:NORMal]:MATRix:PAGE

Sets or queries the displayed page of the numeric display in matrix display mode.

Syntax
```
:DISPlay:NUMeric[:NORMal]:MATRix:
PAGE {<NRf>}
:DISPlay:NUMeric[:NORMal]:MATRix:PAGE?
```
<NRf> = 1 to 9 (page number)

Example
```
:DISPLAY:NUMERIC:NORMAL:MATRIX:
PAGE 1
:DISPLAY:NUMERIC:NORMAL:MATRIX:
PAGE?
-> :DISPLAY:NUMERIC:NORMAL:MATRIX:
PAGE 1
```

Description When the page number is set, the cursor position moves to the beginning of the page.

:DISPlay:NUMeric[:NORMal]:MATRix:PRESet

Presets the display order pattern of displayed items on the numeric display in matrix display mode.

Syntax
```
:DISPlay:NUMeric[:NORMal]:MATRix:
PRESet {<NRf>|EORigin|FORigin|
CLRPage|CLRAll}
```
<NRf> = 1 or EORigin (element reference reset pattern; Element Origin)
<NRf> = 2 or FORigin (function reference reset pattern; Function Origin)
<NRf> = 3 or CLRPage (clear the display items of the current page; Clear Current Page)
<NRf> = 4 or CLRAll (clear the display items of all pages; Clear All Pages)

Example
```
:DISPLAY:NUMERIC:NORMAL:MATRIX:
PRESET 1
:DISPLAY:NUMERIC:NORMAL:MATRIX:
PRESET EORIGIN
```

Description The numeric display item display pattern (order) will be the same as the order when the displayed items are reset using the ITEM setup menu that is displayed on the screen of this instrument (Reset Items Exec). For details on the display pattern that appears when the displayed items are reset, see the features guide, IM WT1801R-01EN.

:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}?

Queries all numeric display settings in 4 Items, 8 Items, or 16 Items display mode.

Syntax
```
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|
VAL16}?
```

:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}:CURSor

Sets or queries the cursor position on the numeric display in 4 Items, 8 Items, or 16 Items display mode.

Syntax
```
:DISPlay:NUMeric[:NORMal]:
{VAL4|VAL8|VAL16}:CURSor {<NRf>}
:DISPlay:NUMeric[:NORMal]:
{VAL4|VAL8|VAL16}:CURSor?
```
<NRf> = 1 to 48 (item number; when VAL4 is specified)
<NRf> = 1 to 96 (item number; when VAL8 is specified)
<NRf> = 1 to 192 (item number; when VAL16 is specified)

Example
```
:DISPLAY:NUMERIC:NORMAL:VAL4:
CURSOR 1
:DISPLAY:NUMERIC:NORMAL:VAL4:
CURSOR?
-> :DISPLAY:NUMERIC:NORMAL:VAL4:
CURSOR 1
```

Description Use an item number to specify the cursor position.

:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}:ITEM<x>

Sets or queries the function, element, and harmonic order of the specified numeric display item in 4 Items, 8 Items, or 16 Items display mode.

Syntax
```
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|
VAL16}:ITEM<x> {NONE|<Function>
[,<Element>][,<Order>]}
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|
VAL16}:ITEM<x>?
```
<x> = 1 to 48 (item number; when VAL4 is specified)
<x> = 1 to 96 (item number; when VAL8 is specified)
<x> = 1 to 192 (item number; when VAL16 is specified)
NONE = No display item
<Function> = {URMS|IRMS|P|S|Q|...}
<Element> = {<NRf>|SIGMa|SIGMB|SIGMC}(<NRf> = 1 to 6)
<Order> = {TOTal|DC|<NRf>}
(<NRf> = 1 to 500)

Example
```
:DISPLAY:NUMERIC:NORMAL:VAL4:
ITEM1 URMS,1
:DISPLAY:NUMERIC:NORMAL:VAL4:ITEM1?
-> :DISPLAY:NUMERIC:NORMAL:VAL4:
ITEM1 URMS,1
:DISPLAY:NUMERIC:NORMAL:VAL4:
ITEM1 UK,1,1
:DISPLAY:NUMERIC:NORMAL:VAL4:ITEM1?
-> :DISPLAY:NUMERIC:NORMAL:VAL4:
ITEM1 UK,1,1
```

Description
- For information about the options available for <Function>, see “Numeric data functions” in “Function option list” on page 5-44.
- If <Element> is omitted, the element is set to 1.
- If <Order> is omitted, the order is set to TOTal.
- <Element> and <Order> are omitted from responses to functions that do not need them.

:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}:PAGE

Sets or queries the displayed page of the numeric display in 4 Items, 8 Items, or 16 Items display mode.

Syntax
```
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|
VAL16}:PAGE {<NRf>}
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|
VAL16}:PAGE?
```
<NRf> = 1 to 12 (page number)

Example
```
:DISPLAY:NUMERIC:NORMAL:VAL4:PAGE 1
:DISPLAY:NUMERIC:NORMAL:VAL4:PAGE?
-> :DISPLAY:NUMERIC:NORMAL:VAL4:PAGE 1
```

Description When the page number is set, the cursor position moves to the beginning of the page.

:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}:PRESet

Presets the display order pattern of displayed items on the numeric display in 4 Items, 8 Items, or 16 Items display mode.

Syntax
```
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|
VAL16}:PRESet {<NRf>|EORigin|FORigin|
CLRPage|CLRAll}
```
<NRf> = 1 or EORigin (element reference reset pattern; Element Origin)
<NRf> = 2 or FORigin (function reference reset pattern; Function Origin)
<NRf> = 3 or CLRPage (clear the display items of the current page; Clear Current Page)
<NRf> = 4 or CLRAll (clear the display items of all pages; Clear All Pages)

Example
```
:DISPLAY:NUMERIC:NORMAL:VAL4:PRESET 1
:DISPLAY:NUMERIC:NORMAL:VAL4:
PRESET EORIGIN
```

Description The numeric display item display pattern (order) will be the same as the order when the displayed items are reset using the ITEM setup menu that is displayed on the screen of this instrument (Reset Items Exec). For details on the display pattern that appears when the displayed items are reset, see the features guide, IM WT1801R-01EN.

:DISPlay:TRENd?

Queries all trend display settings.

Syntax
```
:DISPlay:TRENd?
```

:DISPlay:TRENd:ALL

Collectively sets the on/off status of all trends.

Syntax
```
:DISPlay:TRENd:ALL {<Boolean>}
```
Example
```
:DISPLAY:TREND:ALL ON
```

:DISPlay:TRENd:CLEar

Clears all trends.

Syntax
```
:DISPlay:TRENd:CLEar
```
Example
```
:DISPLAY:TREND:CLEAR
```

:DISPlay:TRENd:FORMat

Sets or queries the display format of all trends.

Syntax
```
:DISPlay:TRENd:FORMat {SINGle|DUAL|
TRIad|QUAD}
:DISPlay:TRENd:FORMat?
```
Example
```
:DISPLAY:TREND:FORMAT SINGLE
:DISPLAY:TREND:FORMAT?
-> :DISPLAY:TREND:FORMAT SINGLE
```

:DISPlay:TRENd:ITEM<x>?

Queries all settings for the specified trend.

Syntax
```
:DISPlay:TRENd:ITEM<x>?
```
<x> = 1 to 16 (item number)

:DISPlay:TRENd:ITEM<x>[:FUNCtion]

Sets or queries the function, element, and harmonic order of the specified trend item.

Syntax
```
:DISPlay:TRENd:ITEM<x>[:
FUNCtion] {<Function>,<Element>
[,<Order>]}
:DISPlay:TRENd:ITEM<x>:FUNCtion?
```
<x> = 1 to 16 (item number)
<Function> = {URMS|IRMS|P|S|Q|...}
<Element> = {<NRf>|SIGMa|SIGMB|SIGMC}
(<NRf> = 1 to 6)
<Order> = {TOTal|DC|<NRf>}
(<NRf> = 1 to 500)

Example
```
:DISPLAY:TREND:ITEM1:FUNCTION URMS,1
:DISPLAY:TREND:ITEM1:FUNCTION?
-> :DISPLAY:TREND:ITEM1:
FUNCTION URMS,1
:DISPLAY:TREND:ITEM1:FUNCTION UK,1,1
:DISPLAY:TREND:ITEM1:FUNCTION?
-> :DISPLAY:TREND:ITEM1:
FUNCTION UK,1,1
```

Description
- For information about the options available for <Function>, see “Numeric data functions” in “Function option list” on page 5-44.
- If <Element> is omitted, the element is set to 1.
- If <Order> is omitted, the order is set to TOTal.
- <Element> and <Order> are omitted from responses to functions that do not need them.

:DISPlay:TRENd:ITEM<x>:SCALing?

Queries all scaling settings for the specified trend.

Syntax
```
:DISPlay:TRENd:ITEM<x>:SCALing?
```
<x> = 1 to 16 (item number)

:DISPlay:TRENd:ITEM<x>:SCALing:MODE

Sets or queries the scaling mode of the specified trend.

Syntax
```
:DISPlay:TRENd:ITEM<x>:SCALing:
MODE {AUTO|MANual}
:DISPlay:TRENd:ITEM<x>:SCALing:MODE?
```
<x> = 1 to 16 (item number)

Example
```
:DISPLAY:TREND:ITEM1:SCALING:
MODE AUTO
:DISPLAY:TREND:ITEM1:SCALING:MODE?
-> :DISPLAY:TREND:ITEM1:SCALING:
MODE AUTO
```

:DISPlay:TRENd:ITEM<x>:SCALing:VALue

Sets or queries the upper and lower limits of the manual scaling of the specified trend.

Syntax
```
:DISPlay:TRENd:ITEM<x>:SCALing:
VALue {<NRf>,<NRf>}
:DISPlay:TRENd:ITEM<x>:SCALing:VALue?
```
<x> = 1 to 16 (item number)
<NRf> = –9.999E+12 to 9.999E+12

Example
```
:DISPLAY:TREND:ITEM1:SCALING:
VALUE 100,-100
:DISPLAY:TREND:ITEM1:SCALING:VALUE?
-> :DISPLAY:TREND:ITEM1:SCALING:
VALUE 100.0E+00,-100.0E+00
```

Description
- Set the upper limit and then the lower limit.
- This command is valid when the scaling mode of the trend (:DISPlay:TRENd:ITEM<x>:SCALing:MODE) is set to MANual.

:DISPlay:TRENd:T<x>

Sets or queries the on/off status of the specified trend.

Syntax
```
:DISPlay:TRENd:T<x> {<Boolean>}
:DISPlay:TRENd:T<x>?
```
<x> = 1 to 16 (item number)

Example
```
:DISPLAY:TREND:T1 ON
:DISPLAY:TREND:T1?
-> :DISPLAY:TREND:T1 1
```

:DISPlay:TRENd:TDIV

Sets or queries the trend horizontal axis (T/div).

Syntax
```
:DISPlay:TRENd:TDIV {<NRf>,<NRf>,
<NRf>}
:DISPlay:TRENd:TDIV?
```
{<NRf>,<NRf>,<NRf>} = 0, 0, 3 to 24, 0, 0
First <NRf> = 1, 3, 6, 12, 24 (hours)
Second <NRf> = 1, 3, 6, 10, 30 (minutes)
Third <NRf> = 3, 6, 10, 30 (seconds)

Example
```
:DISPLAY:TREND:TDIV 0,0,3
:DISPLAY:TREND:TDIV?
-> :DISPLAY:TREND:TDIV 0,0,3
```

Description Set the three <NRf>’s so that one <NRf> is a non-zero value and the other two are zero.

:DISPlay:VECTor?

Queries all vector display settings.

Syntax :DISPlay:VECTor?

Description The vector display is only available on models with the harmonic measurement (/G5 or /G6) option.

:DISPlay:VECTor:FORMat

Sets or queries the display format of all vectors.

Syntax :DISPlay:VECTor:FORMat {SINGle|DUAL}
:DISPlay:VECTor:FORMat?

Example :DISPLAY:VECTOR:FORMAT SINGLE
:DISPLAY:VECTOR:FORMAT?
-> :DISPLAY:VECTOR:FORMAT SINGLE

:DISPlay:VECTor:ITEM<x>?

Queries all settings for the specified vector.

Syntax :DISPlay:VECTor:ITEM<x>?
<x> = 1 or 2 (item number)

:DISPlay:VECTor:ITEM<x>:OBJect

Sets or queries the wiring unit that is displayed using the specified vector.

Syntax :DISPlay:VECTor:ITEM<x>:
OBJect {<Element>}
:DISPlay:VECTor:ITEM<x>:OBJect?
<x> = 1 or 2 (item number)
<Element> = {<NRf>|SIGMa|SIGMB|SIGMC}
(<NRf> = 1 to 6)

Example :DISPLAY:VECTOR:ITEM1:OBJECT SIGMA
:DISPLAY:VECTOR:ITEM1:OBJECT?
-> :DISPLAY:VECTOR:ITEM1:OBJECT SIGMA

:DISPlay:VECTor:ITEM<x>:{UMAG|IMAG}

Sets or queries the voltage or current zoom factor for the vector display.

Syntax :DISPlay:VECTor:ITEM<x>:{UMAG|
IMAG} {<NRf>}
:DISPlay:VECTor:ITEM<x>:{UMAG|IMAG}?
<x> = 1 or 2 (item number)
<NRf> = 0.100 to 100.000

Example :DISPLAY:VECTOR:ITEM1:UMAG 1
:DISPLAY:VECTOR:ITEM1:UMAG?
-> :DISPLAY:VECTOR:ITEM1:UMAG 1.000

:DISPlay:VECTor:NUMeric

Sets or queries the on/off status of the numeric data display on the vector display.

Syntax :DISPlay:VECTor:NUMeric {<Boolean>}
:DISPlay:VECTor:NUMeric?

Example :DISPLAY:VECTOR:NUMERIC ON
:DISPLAY:VECTOR:NUMERIC?
-> :DISPLAY:VECTOR:NUMERIC 1

:DISPlay:WAVE?

Queries all waveform display settings.

Syntax :DISPlay:WAVE?

:DISPlay:WAVE:ALL

Collectively sets the on/off status of all waveform displays.

Syntax :DISPlay:WAVE:ALL {<Boolean>}

Example :DISPLAY:WAVE:ALL ON

:DISPlay:WAVE:FORMat

Sets or queries the display format of all waveforms.

Syntax :DISPlay:WAVE:FORMat {SINGle|DUAL|
TRIad|QUAD|HEXa}
:DISPlay:WAVE:FORMat?

Example :DISPLAY:WAVE:FORMAT SINGLE
:DISPLAY:WAVE:FORMAT?
-> :DISPLAY:WAVE:FORMAT SINGLE

:DISPlay:WAVE:GRATicule

Sets or queries the graticule (grid) type.

Syntax :DISPlay:WAVE:GRATicule {GRID|FRAMe|
CROSshair}
:DISPlay:WAVE:GRATicule?

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

:DISPlay:WAVE:INTerpolate

Sets or queries the waveform interpolation method.

Syntax :DISPlay:WAVE:INTerpolate {OFF|LINE}
:DISPlay:WAVE:INTerpolate?

Example :DISPLAY:WAVE:INTERPOLATE LINE
:DISPLAY:WAVE:INTERPOLATE?
-> :DISPLAY:WAVE:INTERPOLATE LINE

:DISPlay:WAVE:MAPPing?

Queries all split screen waveform mapping settings.

Syntax :DISPlay:WAVE:MAPPing?

:DISPlay:WAVE:MAPPing[:MODE]

Sets or queries the split screen waveform mapping mode.

Syntax :DISPlay:WAVE:MAPPing[:MODE] {AUTO|
FIXed|USER}
:DISPlay:WAVE:MAPPing:MODE?

Example :DISPLAY:WAVE:MAPPING:MODE AUTO
:DISPLAY:WAVE:MAPPING:MODE?
-> :DISPLAY:WAVE:MAPPING:MODE AUTO

:DISPlay:WAVE:MAPPing:{U<x>|I<x>|SPEed|TORQue|AUX<x>}

Sets or queries the split screen voltage, current, rotating speed, torque, or auxiliary signal waveform mapping setting.

Syntax
```
:DISPlay:WAVE:MAPPing:{U<x>|I<x>|
SPEed|TORQue|AUX<x>} {<NRf>}
:DISPlay:WAVE:MAPPing:{U<x>|I<x>|
SPEed|TORQue|AUX<x>}?
```
U<x> and I<x>'s <x> = 1 to 6 (element)
AUX<x>'s <x> = 1 or 2 (AUX input channel)
<NRf> = 0 to 5

Example
```
:DISPLAY:WAVE:MAPPING:U1 0
:DISPLAY:WAVE:MAPPING:U1?
-> :DISPLAY:WAVE:MAPPING:U1 0
```

Description
- This command is valid when the waveform mapping method (:DISPlay:WAVE:MAPPing[:MODE]) is set to USER.
- SPEed and TORQue are only valid on models with the motor evaluation function (/MTR) option.
- AUX<x> is only valid on models with the auxiliary input (/AUX) option.

:DISPlay:WAVE:POSition?

Queries all waveform vertical position (center position level) settings.

Syntax
```
:DISPlay:WAVE:POSition?
```

:DISPlay:WAVE:POSition:{U<x>|I<x>}

Sets or queries the vertical position (center position level) of the specified element's voltage or current waveform.

Syntax
```
:DISPlay:WAVE:POSition:{U<x>|
I<x>} {<NRf>}
:DISPlay:WAVE:POSition:{U<x>|I<x>}?
```
<x> = 1 to 6 (element)
<NRf> = –130.000 to 130.000 (%)

Example
```
:DISPLAY:WAVE:POSITION:U1 0
:DISPLAY:WAVE:POSITION:U1?
-> :DISPLAY:WAVE:POSITION:U1 0.000
```

:DISPlay:WAVE:POSition:{UALL|IALL}

Collectively sets the vertical positions (center position levels) of the voltage or current waveforms of all elements.

Syntax
```
:DISPlay:WAVE:POSition:{UALL|
IALL} {<NRf>}
```
<NRf> = –130.000 to 130.000 (%)

Example
```
:DISPLAY:WAVE:POSITION:UALL 0
```

:DISPlay:WAVE:SVALue

Sets or queries the on/off status of the scale value display.

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

Example
```
:DISPLAY:WAVE:SVALUE ON
:DISPLAY:WAVE:SVALUE?
-> :DISPLAY:WAVE:SVALUE 1
```

:DISPlay:WAVE:TDIV

Sets or queries the waveform Time/div value.

Syntax
```
:DISPlay:WAVE:TDIV {<Time>}
:DISPlay:WAVE:TDIV?
```
<Time> = 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500 (ms), 1, 2 (s)

Example
```
:DISPLAY:WAVE:TDIV 5MS
:DISPLAY:WAVE:TDIV?
-> :DISPLAY:WAVE:TDIV 5.00E-03
```

Description The specifiable Time/div value is up to 1/10 of the data update interval (:RATE).

:DISPlay:WAVE:TLABel

Sets or queries the on/off status of the waveform labels.

Syntax
```
:DISPlay:WAVE:TLABel {<Boolean>}
:DISPlay:WAVE:TLABel?
```

Example
```
:DISPLAY:WAVE:TLABEL OFF
:DISPLAY:WAVE:TLABEL?
-> :DISPLAY:WAVE:TLABEL 0
```

:DISPlay:WAVE:TRIGger?

Queries all trigger settings.

Syntax
```
:DISPlay:WAVE:TRIGger?
```

:DISPlay:WAVE:TRIGger:LEVel

Sets or queries the trigger level.

Syntax
```
:DISPlay:WAVE:TRIGger:LEVel {<NRf>}
:DISPlay:WAVE:TRIGger:LEVel?
```
<NRf> = –100.0 to 100.0 (%)

Example
```
:DISPLAY:WAVE:TRIGGER:LEVEL 0
:DISPLAY:WAVE:TRIGGER:LEVEL?
-> :DISPLAY:WAVE:TRIGGER:LEVEL 0.0
```

Description Set the value in terms of a percentage of the full scale value displayed on the screen.

:DISPlay:WAVE:TRIGger:MODE

Sets or queries the trigger mode.

Syntax
```
:DISPlay:WAVE:TRIGger:MODE {AUTO|
NORMal|OFF}
:DISPlay:WAVE:TRIGger:MODE?
```

Example
```
:DISPLAY:WAVE:TRIGGER:MODE AUTO
:DISPLAY:WAVE:TRIGGER:MODE?
-> :DISPLAY:WAVE:TRIGGER:MODE AUTO
```

Description If the waveform display is enabled and the trigger mode is set to Auto or Normal, the data update interval depends on the trigger operation.

:DISPlay:WAVE:TRIGger:SLOPe

Sets or queries the trigger slope.

Syntax
```
:DISPlay:WAVE:TRIGger:SLOPe {RISE|
FALL|BOTH}
:DISPlay:WAVE:TRIGger:SLOPe?
```

Example
```
:DISPLAY:WAVE:TRIGGER:SLOPE RISE
:DISPLAY:WAVE:TRIGGER:SLOPE?
-> :DISPLAY:WAVE:TRIGGER:SLOPE RISE
```

:DISPlay:WAVE:TRIGger:SOURce

Sets or queries the trigger source.

Syntax `:DISPlay:WAVE:TRIGger:SOURce {U<x>|I<x>|EXTernal}`
`:DISPlay:WAVE:TRIGger:SOURce?`
<x> = 1 to 6 (element)
EXTernal = External trigger input (Ext Clk)

Example
```
:DISPLAY:WAVE:TRIGGER:SOURCE U1
:DISPLAY:WAVE:TRIGGER:SOURCE?
-> :DISPLAY:WAVE:TRIGGER:SOURCE U1
```

:DISPlay:WAVE:{U<x>|I<x>|SPEed|TORQue|AUX<x>}

Sets or queries the on/off status of the voltage, current, rotating speed, torque, or auxiliary signal waveform display.

Syntax `:DISPlay:WAVE:{U<x>|I<x>|SPEed|TORQue|AUX<x>} {<Boolean>}`
`:DISPlay:WAVE:{U<x>|I<x>|SPEed|TORQue|AUX<x>}?`
U<x> and I<x>'s <x> = 1 to 6 (element)
AUX<x>'s <x> = 1 or 2 (AUX input channel)

Example
```
:DISPLAY:WAVE:U1 ON
:DISPLAY:WAVE:U1?
-> :DISPLAY:WAVE:U1 1
```

Description
- SPEed and TORQue are only valid on models with the motor evaluation function (/MTR) option.
- AUX<x> is only valid on models with the auxiliary input (/AUX) option.

:DISPlay:WAVE:VZoom?

Queries all waveform vertical zoom factor settings.

Syntax `:DISPlay:WAVE:VZoom?`

:DISPlay:WAVE:VZoom:{U<x>|I<x>}

Sets or queries the vertical zoom factor of the specified element's voltage or current waveform.

Syntax `:DISPlay:WAVE:VZoom:{U<x>|I<x>} {<NRf>}`
`:DISPlay:WAVE:VZoom:{U<x>|I<x>}?`
<x> = 1 to 6 (element)
<NRf> = 0.1 to 100

Example
```
:DISPLAY:WAVE:VZOOM:U1 1
:DISPLAY:WAVE:VZOOM:U1?
-> :DISPLAY:WAVE:VZOOM:U1 1.00
```

Description For details on the available zoom factors, see the features guide, IM WT1801R-01EN.

:DISPlay:WAVE:VZoom:{UALL|IALL}

Collectively sets the vertical zoom factor for the voltage or current waveforms of all elements.

Syntax `:DISPlay:WAVE:VZoom:{UALL|IALL} {<NRf>}`
<NRf> = 0.1 to 100

Example `:DISPLAY:WAVE:VZOOM:UALL 1`

Description For details on the available zoom factors, see the features guide, IM WT1801R-01EN.

Function option list (settings that can be used for <Function>)

- Numeric data functions
- Numeric list data functions

Numeric data functions

Applicable commands

```
:AOUTput[:NORMal]:CHANnel<x> {NONE|<Function>[,<Element>][,<Order>]}
:DISPlay:NUMeric:CUSTom:ITEM<x>[:FUNCtion] {<Function>[,<Element>]
  [,<Order>]|<String>}
:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}:ITEM<x> {NONE|<Function>
  [,<Element>][,<Order>]}
:DISPlay:NUMeric[:NORMal]:MATRix:ITEM<x> {NONE|<Function>[,<Element>]
  [,<Order>]}
:DISPlay:NUMeric[:NORMal]:ALL:CURSor {<Function>}
:DISPlay:TRENd:ITEM<x>[:FUNCtion] {<Function>,<Element>[,<Order>]}
:FILE:SAVE:NUMeric:NORMal:<Function> {<Boolean>}
:MEASure:EVENt<x>:EXPRession:ITEM {<Function>[,<Element>][,<Order>]}
:NUMeric[:NORMal]:ITEM<x> {NONE|<Function>[,<Element>][,<Order>]}
:STORe:NUMeric[:NORMal]:<Function> {<Boolean>}
```

<Function>	Function name used on the menu (Numeric display header name)	<Element>	<Order>
URMS	Urms	Required	Not required
UMN	Umn	Required	Not required
UDC	Udc	Required	Not required
URMN	Urmn	Required	Not required
UAC	Uac	Required	Not required
IRMS	Irms	Required	Not required
IMN	Imn	Required	Not required
IDC	Idc	Required	Not required
IRMN	Irmn	Required	Not required
IAC	Iac	Required	Not required
P	P	Required	Not required
S	S	Required	Not required
Q	Q	Required	Not required
LAMBda	λ	Required	Not required
PHI	φ	Required	Not required
FU	FreqU(fU)	Required	Not required
FI	FreqI(fI)	Required	Not required
UPPeak	U+peak(U+pk)	Required	Not required
UMPeak	U-peak(U-pk)	Required	Not required
IPPeak	I+peak(I+pk)	Required	Not required
IMPeak	I-peak(I-pk)	Required	Not required
CFU	CfU	Required	Not required
CFI	CfI	Required	Not required
PC	Pc	Required	Not required
PPPeak	P+peak(P+pk)	Required	Not required
PMPeak	P-peak(P-pk)	Required	Not required
TIME	Time	Required	Not required
WH	WP	Required	Not required
WHP	WP+	Required	Not required

<Function>	Function name used on the menu (Numeric display header name)	<Element>	<Order>
WHM	WP-	Required	Not required
AH	q	Required	Not required
AHP	q+	Required	Not required
AHM	q-	Required	Not required
WS	WS	Required	Not required
WQ	WQ	Required	Not required
ETA1 to ETA4	η1 to η4	Not required	Not required
F1 to F20	F1 to F20	Not required	Not required
EV1 to EV8	Event1 to Event8	Not required	Not required
Functions that require the harmonic measurement (/G5 or /G6) option			
UK	U(k)	Required	Required
IK	I(k)	Required	Required
PK	P(k)	Required	Required
SK	S(k)	Required	Required
QK	Q(k)	Required	Required
LAMBDAK	λ(k)	Required	Required
PHIK	φ(k)	Required	Required
PHIUk	φU(k)	Required	Required
PHIIk	φI(k)	Required	Required
Zk	Z(k)	Required	Required
RSk	Rs(k)	Required	Required
XSk	Xs(k)	Required	Required
RPk	Rp(k)	Required	Required
XPk	Xp(k)	Required	Required
UHDFk	Uhdf(k)	Required	Required
IHDFk	Ihdf(k)	Required	Required
PHDFk	Phdf(k)	Required	Required
UTHD	Uthd	Required	Not required
ITHD	Ithd	Required	Not required
PTHD	Pthd	Required	Not required
UTHF	Uthf	Required	Not required
ITHF	Ithf	Required	Not required
UTIF	Utif	Required	Not required
ITIF	Itif	Required	Not required
HVF	hvf	Required	Not required
HCF	hcf	Required	Not required
KFACtor	K-factor	Required	Not required
PHI_U1U2	φUi-Uj	Required	Not required
PHI_U1U3	φUi-Uk	Required	Not required
PHI_U1I1	φUi-Ii	Required	Not required
PHI_U2I2	φUj-Ij	Required	Not required
PHI_U3I3	φUj-Ik	Required	Not required
FPLL1	fPLL1	Not required	Not required
FPLL2	fPLL2	Not required	Not required
Functions of the delta computation			
DU1	ΔU1	Required (only Σ)	Not required
DU2	ΔU2	Required (only Σ)	Not required
DU3	ΔU3	Required (only Σ)	Not required
DUS	ΔUΣ	Required (only Σ)	Not required
DI	ΔI	Required (only Σ)	Not required
DP1	ΔP1	Required (only Σ)	Not required
DP2	ΔP2	Required (only Σ)	Not required
DP3	ΔP3	Required (only Σ)	Not required
DPS	ΔPΣ	Required (only Σ)	Not required

<Function>	Function name used on the menu (Numeric display header name)	<Element>	<Order>
Functions that require the motor evaluation function (/MTR) option			
SPEed	Speed	Not required	Not required
TORQue	Torque	Not required	Not required
SYNCsp	SyncSp	Not required	Not required
SLIP	Slip	Not required	Not required
PM	Pm	Not required	Not required
EAU	EaU	Required	Not required
EAI	EaI	Required	Not required
Functions that require the auxiliary input (/AUX) option			
AUX1	Aux1	Not required	Not required
AUX2	Aux2	Not required	Not required

Note

- For functions in the list above that do not require the element to be specified but whose commands have a parameter for specifying the element (<Element>), omit the parameter or set it to 1.
- Likewise, for functions in the list above that do not require the harmonic order to be specified but whose commands have a parameter for specifying the harmonic order (<Order>), omit the parameter or set it to TOTal.

Applicable commands

`:NUMeric[:NORMal]:ITEM<x> {NONE|`**`<Function>`**`[,<Element>][,<Order>]}`

For the above commands, the following functions can also be selected.

These functions are not displayed on the menu.

<Function>	Description	<Element>	<Order>
URANge	Voltage range	Required	Not required
IRANge	Current range	Required	Not required
SPDRange	Rotating speed input range	Not required	Not required
TRQRange	Torque input range	Not required	Not required
AUXRange	Auxiliary input range	Required (only 1 or 2)	Not required
UNPower	Input element data update status for normal measurement	Not required	Not required
UNMotor	Motor evaluation input data update status for normal measurement	Not required	Not required
UNAux	Auxiliary signal input data update status for normal measurement	Not required	Not required
UHPower	Input element data update status for harmonic measurement	Not required	Not required
UWCHannel	Data update status of waveform measurement	Not required	Not required

For information about data update status, see section appendix 10 in the features guide, IM WT1801R-01EN.

Numeric list data functions

These functions require the harmonic measurement option.

Applicable commands

:DISPlay:BAR:ITEM<x>[:FUNCtion] {**<Function>**,<Element>}

:DISPlay:NUMeric[:NORMal]:LIST:ITEM<x> {**<Function>**,<Element>}

:NUMeric:LIST:ITEM<x> {NONE|**<Function>**,<Element>}

<Function>	Function name used on the menu
U	U
I	I
P	P
S	S
Q	Q
LAMBda	λ
PHI	φ
PHIU	φU
PHII	φI
Z	Z
RS	Rs
XS	Xs
RP	Rp
XP	Xp
The function options listed below are only valid with :NUMeric:LIST:ITEM<x>.	
UHDF	Uhdf
IHDF	Ihdf
PHDF	Phdf

5.7 FILE Group

The commands in this group deal with file operations.
You can perform the same operations and make the same settings and queries that you can make by pressing FILE on the front panel.

* The commands in the FILE group use a dedicated file list, and this list is not synchronized to the file list used through the menus.

:FILE?

Queries all file operation settings.

Syntax :FILE?

:FILE:CDIRectory

Changes the current directory.

Syntax :FILE:CDIRectory {<String>}
<String> = Directory name
Example :FILE:CDIRECTORY "TEST"
Description Specify ".." to move up to the parent directory.

:FILE:DELete:IMAGe:{BMP|PNG|JPEG}

Deletes the specified screen image data file.

Syntax :FILE:DELete:IMAGe:
{BMP|PNG|JPEG} {<String>}
<String> = File name
Example :FILE:DELETE:IMAGE:BMP "IMAGE1"
Description Specify the file name without an extension.

:FILE:DELete:NUMeric:ASCii

Deletes the specified numeric data file.

Syntax :FILE:DELete:NUMeric:ASCii {<String>}
<String> = File name
Example :FILE:DELETE:NUMERIC:ASCII "NUM1"
Description Specify the file name without an extension.

:FILE:DELete:SETup

Deletes the specified setup parameter file.

Syntax :FILE:DELete:SETup {<String>}
<String> = File name
Example :FILE:DELETE:SETUP "SETUP1"
Description Specify the file name without an extension.

:FILE:DELete:STORe:{DATA|HEADer}

Deletes the specified stored numeric data file.

Syntax :FILE:DELete:STORe:{DATA|
HEADer} {<String>}
<String> = File name
Example :FILE:DELETE:STORE:DATA "STR1"
Description Specify the file name without an extension.

:FILE:DELete:WAVE:ASCii

Deletes the specified waveform display data file.

Syntax :FILE:DELete:WAVE:ASCii {<String>}
<String> = File name
Example :FILE:DELETE:WAVE:ASCII "WAVE1"
Description Specify the file name without an extension.

:FILE:DRIVe

Sets the current drive.

Syntax :FILE:DRIVe {USER|RAM|USB[,<NRf>]|
NETWork}
USER = Internal memory drive
RAM = Internal RAM disk
USB = USB memory device drive
<NRf> = 0 or 1 (drive number)
NETWork = Network drive
Example :FILE:DRIVE USER

:FILE:FILTer

Sets or queries the file list filter.

Syntax :FILE:FILTer {ALL|ITEM}
:FILE:FILTer?
Example :FILE:FILTER ALL
:FILE:FILTER?
-> :FILE:FILTER ALL

:FILE:FREE?

Queries the free space (in bytes) on the current drive.

Syntax :FILE:FREE?
Example :FILE:FREE? -> 20912128

:FILE:LOAD:ABORt

Aborts a file loading operation.

Syntax :FILE:LOAD:ABORt
Example :FILE:LOAD:ABORT

:FILE:LOAD:SETup

Loads the specified setup parameter file.

Syntax :FILE:LOAD:SETup {<String>}
<String> = File name
Example :FILE:LOAD:SETUP "SETUP1"
Description • Specify the file name without an extension.
• This command is an overlap command.

:FILE:PATH?

Queries the absolute path of the current directory.

Syntax :FILE:PATH?

Example :FILE:PATH? -> "USB-0/TEST"

:FILE:SAVE?

Queries all file save settings.

Syntax :FILE:SAVE?

:FILE:SAVE:ABORt

Aborts a file saving operation.

Syntax :FILE:SAVE:ABORt

Example :FILE:SAVE:ABORT

:FILE:SAVE:ANAMing

Sets or queries the auto naming feature for saving files.

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

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

:FILE:SAVE:COMMent

Sets or queries the comment that will be added to files that are saved.

Syntax :FILE:SAVE:COMMent {<String>}
:FILE:SAVE:COMMent?
<String> = Up to 30 characters

Example :FILE:SAVE:COMMENT "CASE1"
:FILE:SAVE:COMMENT?
-> :FILE:SAVE:COMMENT "CASE1"

:FILE:SAVE:NUMeric[:EXECute]

Saves numeric data to a file.

Syntax :FILE:SAVE:NUMeric[:EXECute] {<String>}
<String> = File name

Example :FILE:SAVE:NUMERIC:EXECUTE "NUM1"

Description
- Specify the file name without an extension.
- This command is an overlap command.

:FILE:SAVE:NUMeric:ITEM

Sets or queries the method that is used to select which items are saved when numeric data is saved to a file.

Syntax :FILE:SAVE:NUMeric:ITEM {DISPlayed|SELected}
:FILE:SAVE:NUMeric:ITEM?
DISPlayed = Automatic selection method in which all the items that are displayed on the screen are selected
SELected = Manual selection method

Example :FILE:SAVE:NUMERIC:ITEM SELECTED
:FILE:SAVE:NUMERIC:ITEM?
-> :FILE:SAVE:NUMERIC:ITEM SELECTED

Description The available options are explained below.
DISPlayed = The numeric items that are displayed on the screen are saved to the file.
SELected = The numeric items that are specified with the commands that start with ":FILE:SAVE:NUMeric:NORMal: . . ." are saved to the file.

:FILE:SAVE:NUMeric:NORMal?

Queries all numeric data file save settings (for the manual save item selection method).

Syntax :FILE:SAVE:NUMeric:NORMal?

Description This command is valid when the save item selection method (:FILE:SAVE:NUMeric:ITEM) is set to SELected (the manual selection method).

:FILE:SAVE:NUMeric:NORMal:ALL

Collectively sets the on/off status of the output of all element functions when numeric data is saved to a file.

Syntax :FILE:SAVE:NUMeric:NORMal:ALL {<Boolean>}

Example :FILE:SAVE:NUMERIC:NORMAL:ALL ON

:FILE:SAVE:NUMeric:NORMal:{ELEMent<x>|SIGMA|SIGMB|SIGMC}

Sets or queries the on/off status of the output of the specified element or wiring unit ΣA, ΣB, or ΣC when numeric data is saved to a file.

Syntax :FILE:SAVE:NUMeric:NORMal:{ELEMent<x>|SIGMA|SIGMB|SIGMC} {<Boolean>}
:FILE:SAVE:NUMeric:NORMal:{ELEMent<x>|SIGMA|SIGMB|SIGMC}?
<x> = 1 to 6

Example :FILE:SAVE:NUMERIC:NORMAL:ELEMENT1 ON
:FILE:SAVE:NUMERIC:NORMAL:ELEMENT1?
-> :FILE:SAVE:NUMERIC:NORMAL:ELEMENT1 1

:FILE:SAVE:NUMeric:NORMal:<Function>

Sets or queries the on/off status of the specified function's output when numeric data is saved to a file.

Syntax
```
:FILE:SAVE:NUMeric:NORMal:
<Function> {<Boolean>}
:FILE:SAVE:NUMeric:NORMal:<Function>?
```
<Function> = {URMS|IRMS|P|S|Q|...}

Example
```
:FILE:SAVE:NUMERIC:NORMAL:URMS ON
:FILE:SAVE:NUMERIC:NORMAL:URMS?
-> :FILE:SAVE:NUMERIC:NORMAL:URMS 1
```

Description For information about the options available for <Function>, see "Numeric data functions" in "Function option list" at the end of the DISPlay group on page 5-44.

:FILE:SAVE:NUMeric:NORMal:PRESet<x>

Presets the output on/off pattern of the element functions to be used when numeric data is saved to a file.

Syntax
```
:FILE:SAVE:NUMeric:NORMal:PRESet<x>
```
<x> = 1 or 2 (preset number)

Example
```
:FILE:SAVE:NUMERIC:NORMAL:PRESET1
```

Description For details on the output setting patterns that result when the pattern is reset, see the features guide, IM WT1801R-01EN.

:FILE:SAVE:SETup[:EXECute]

Saves setup parameters to a file.

Syntax
```
:FILE:SAVE:SETup[:EXECute] {<String>}
```
<String> = File name

Example
```
:FILE:SAVE:SETUP:EXECUTE "SETUP1"
```

Description
- Specify the file name without an extension.
- This command is an overlap command.

:FILE:SAVE:WAVE[:EXECute]

Saves waveform display data to a file.

Syntax
```
:FILE:SAVE:WAVE[:EXECute] {<String>}
```
<String> = File name

Example
```
:FILE:SAVE:WAVE:EXECUTE "WAVE1"
```

Description
- Specify the file name without an extension.
- This command is an overlap command.

5.8 HARMonics Group

The commands in this group deal with harmonic measurement.
You can make the same settings and queries that you can make by pressing HRM SET on the front panel.
The commands in this group are only valid on models with the simultaneous dual harmonic measurement (/G6) option or the harmonic measurement (/G5) option.

:HARMonics<x>?

Queries all harmonic measurement settings.

Syntax :HARMonics<x>?
<x> = 1 or 2 (harmonic measurement group)

Description "HARMonics2" is only valid on models with the simultaneous dual harmonic measurement (/G6) option.

:HARMonics<x>:CONFigure?

Queries the harmonic measurement groups of all elements.

Syntax :HARMonics<x>:CONFigure?

Description
- This command is only valid on models with the simultaneous dual harmonic measurement (/G6) option.
- The <x> value in HARMonics<x> has no meaning in the query.

:HARMonics<x>:CONFigure[:ALL]

Collectively sets the harmonic measurement group of all elements.

Syntax :HARMonics<x>:CONFigure[:ALL] {<NRf>}
<NRf> = 1 (Hrm1), 2 (Hrm2)

Example :HARMONICS:CONFIGURE:ALL 1

Description
- This command is only valid on models with the simultaneous dual harmonic measurement (/G6) option.
- The <x> value in HARMonics<x> has no meaning in the setting.

:HARMonics<x>:CONFigure:ELEMent<x>

Sets or queries the harmonic measurement group of the specified element.

Syntax :HARMonics<x>:CONFigure:
ELEMent<x> {<NRf>}
:HARMonics<x>:CONFigure:ELEMent<x>?
ELEMent<x>'s <x> = 1 to 6 (element)
<NRf> = 1 (Hrm1), 2 (Hrm2)

Example :HARMONICS:CONFIGURE:ELEMENT1 1
:HARMONICS:CONFIGURE:ELEMENT1?
-> :HARMONICS1:CONFIGURE:ELEMENT1 1

Description
- This command is only valid on models with the simultaneous dual harmonic measurement (/G6) option.
- The <x> value in HARMonics<x> has no meaning in the setting or query.

:HARMonics<x>:CONFigure:{SIGMA|SIGMB|SIGMC}

Collectively sets the harmonic measurement group of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).

Syntax :HARMonics<x>:CONFigure:
{SIGMA|SIGMB|SIGMC} {<NRf>}
<NRf> = 1 (Hrm1), 2 (Hrm2)

Example :HARMONICS:CONFIGURE:SIGMA 1

Description
- This command is only valid on models with the simultaneous dual harmonic measurement (/G6) option.
- The <x> value in HARMonics<x> has no meaning in the setting.

:HARMonics<x>:ORDer

Sets or queries the maximum and minimum harmonic orders that are analyzed.

Syntax :HARMonics<x>:ORDer {<NRf>,<NRf>}
:HARMonics<x>:ORDer?
<x> = 1 or 2 (harmonic measurement group)
First <NRf> = 0 or 1 (minimum harmonic order that is analyzed)
Second <NRf> = 1 to 500 (maximum harmonic order that is analyzed)

Example :HARMONICS:ORDER 1,100
:HARMONICS:ORDER?
-> :HARMONICS1:ORDER 1,100

Description When the data update interval is Auto, only the "HARMonics1" is valid even on models with the simultaneous dual harmonic measurement (/G6) option .

:HARMonics<x>:PLLSource

Sets or queries the PLL source.

Syntax :HARMonics<x>:PLLSource {U<x>|I<x>|
EXTernal}
:HARMonics<x>:PLLSource?
HARMonics<x>'s <x> = 1 or 2 (harmonic measurement group)
U<x> and I<x>'s <x> = 1 to 6 (element)
EXTernal = External clock input (Ext Clk)

Example :HARMONICS:PLLSOURCE U1
:HARMONICS:PLLSOURCE?
-> :HARMONICS1:PLLSOURCE U1

Description When the data update interval is Auto, only the "HARMonics1" is valid even on models with the simultaneous dual harmonic measurement (/G6) option .

:HARMonics<x>:THD

Sets or queries the equation used to compute the THD (total harmonic distortion).

Syntax `:HARMonics<x>:THD {TOTal|FUNDamental}`
`:HARMonics<x>:THD?`
<x> = 1 or 2 (harmonic measurement group)

Example `:HARMONICS:THD TOTAL`
`:HARMONICS:THD?`
`-> :HARMONICS1:THD TOTAL`

Description When the data update interval is Auto, only the "HARMonics1" is valid even on models with the simultaneous dual harmonic measurement (/G6) option .

:HARMonics<x>:POINt

Sets or queries the number of FFT points to use for harmonic measurement.

Syntax `:HARMonics<x>:POINt {<NRf>}`
`:HARMonics<x>:POINt?`
<x> = 1 (harmonic measurement group)

Example `:HARMONICS:POINT 1024`
`:HARMONICS:POINT?`
`-> :HARMONICS1:POINT 1024`

Description
- When the data update interval is not Auto, this is selected automatically, so setting this will not affect the operation.
- When the data update interval is Auto, only the "HARMonics1" is valid even on models with the simultaneous dual harmonic measurement (/G6) option.

5.9 HISTory Group

The commands in this group deal with the history feature.
For details on the history feature, see Appendix 2.

:HISTory?

Queries all history feature settings.

Syntax `:HISTory?`

:HISTory:COMMunicate?

Queries all history communication output settings.

Syntax `:HISTory:COMMunicate?`

:HISTory:COMMunicate:COUNt

Sets or queries the history communication output count.

Syntax `:HISTory:COMMunicate:COUNt {<NRf>}`
`:HISTory:COMMunicate:COUNt?`
<NRf> = 1 to 100

Example `:HISTORY:COMMUNICATE:COUNT 5`
`:HISTORY:COMMUNICATE:COUNT?`
`-> :HISTORY:COMMUNICATE:COUNT 5`

Description
- Manually select the number times to output update data at once during history communication output.
- This setting is valid when the history communication output mode (:HISTory:COMMunicate:OUTPutmode) is set to N update (NUPDate).

:HISTory:COMMunicate:FORMat

Sets or queries the history communication output data format.

Syntax `:HISTory:COMMunicate:FORMat {ASCii|FLOat}`
`:HISTory:COMMunicate:FORMat?`

Example `:HISTORY:COMMUNICATE:FORMAT FLOAT`
`:HISTORY:COMMUNICATE:FORMAT?`
`-> :HISTORY:COMMUNICATE:FORMAT FLOAT`

Description
- The history communication output data is currently fixed to “FLOat.”
- For details on the numeric data output format, see the explanation in “:NUMeric:FORMat.”

:HISTory:COMMunicate:HOLD

Sets or queries the on/off (hold/release) status of the history communication output data hold feature.

Syntax `:HISTory:COMMunicate:HOLD {<Boolean>}`
`:HISTory:COMMunicate:HOLD?`

Example `:HISTORY:COMMUNICATE:HOLD ON`
`:HISTORY:COMMUNICATE:HOLD?`
`-> :HISTORY:COMMUNICATE:HOLD 1`

Description The measurement data in the history is retained for communication output. The amount of retained data is dictated by the number of output counts according to the specified history communication output mode.

:HISTory:COMMunicate:OUTPutmode

Sets or queries the history communication output mode.

Syntax `:HISTory:COMMunicate:OUTPutmode {NUPDate}`
`:HISTory:COMMunicate:OUTPutmode?`
NUPDate = N update mode

Example `:HISTORY:COMMUNICATE:OUTPUTMODE NUPDATE`
`:HISTORY:COMMUNICATE:OUTPUTMODE?`
`-> :HISTORY:COMMUNICATE:OUTPUTMODE NUPDATE`

Description Sets the history communication output’s start position and output count and the extended event status register’s (EESR’s) HST control method.
- N update mode (NUPDate)
  History communication output update complete is indicated at every measured data update, and the history output start position is set automatically. If you want to continuously acquire numeric data whose data update interval is 10 ms, select this mode.

:HISTory:NUMeric:LIST:VALue?

Queries the history communication numeric list data.

Syntax :HISTory:NUMeric:LIST:VALue?

Example :HISTORY:NUMERIC:LIST:VALUE?
-> #N (N-digit byte number)(data byte sequence)

Description
- This command outputs several update intervals of the same item obtained by querying for the harmonic measurement numeric list data (:NUMeric:LIST:VALue?).
- The number of output intervals is determined according to the history communication output mode.
- The output items, the number of output items, and the output format for a single interval are the same as those of ":NUMeric:LIST:VALue?."
- With this command, you cannot specify the parameter (a specific item number) as you can with ":NUMeric:LIST:VALue?."

:HISTory:NUMeric:VALue?

Queries the history communication numeric data.

Syntax :HISTory:NUMeric:VALue?

Example :HISTORY:NUMERIC:VALUE?
-> #N (N-digit byte number)(data byte sequence)

Description
- This command outputs several update intervals of the same item obtained by querying for the numeric data (:NUMeric[:NORMal]:VALue?).
- The number of output intervals is determined according to the history communication output mode.
- The output items, the number of output items, and the output format for a single interval are the same as those of ":NUMeric[:NORMal]:VALue?." For details, see Appendix 2.
- With this command, you cannot specify the parameter (a specific item number) as you can with ":NUMeric[:NORMal]:VALue?."

5.10 HOLD Group

The command in this group deals with the output data hold feature.
You can make the same settings and queries that you can make by pressing HOLD on the front panel.

:HOLD

Sets or queries the on/off status of the output hold feature for display, communication, and other types of data.

Syntax `:HOLD {<Boolean>}`
`:HOLD?`

Example `:HOLD OFF`
`:HOLD? -> :HOLD 0`

5.11 HSPeed Group

The commands in this group deal with the high speed data capturing feature.
These commands allow you to enter and query the same settings that are available under ITEM in the “HS Items”menu and under FORM in the “HS Settings” menu on the front panel.

:HSPeed?

Queries all high speed data capturing feature settings.

Syntax :HSPeed?

:HSPeed:CAPTured?

Queries the number of captures that have been performed in high speed data capturing.

Syntax :HSPeed:CAPTured?
Example :HSPEED:CAPTURED? -> 200
Description Returns the number displayed next to “Captured” in the status display at the top of the screen during capturing.

:HSPeed:COUNt

Sets or queries the number of data captures.

Syntax :HSPeed:COUNt {<NRf>|INFinite}
:HSPeed:COUNt?
<NRf> = 1 to 10000000
INFinite = No limit
Example :HSPEED:COUNT INFINITE
:HSPEED:COUNT?
-> :HSPEED:COUNT INFINITE

:HSPeed:DISPlay?

Queries all display settings of high speed data capturing mode.

Syntax :HSPeed:DISPlay?
Description The “:HSPeed:DISPlay:. . .” commands perform the same settings and queries as the “:DISPlay:HSPeed:. . .” commands.

:HSPeed:DISPlay:COLumn?

Queries all column settings of high speed data capturing mode.

Syntax :HSPeed:DISPlay:COLumn?

:HSPeed:DISPlay:COLumn:ITEM<x>

Sets or queries a column display item of high speed data capturing mode.

Syntax :HSPeed:DISPlay:COLumn:ITEM<x> {NONE|<Element>}
HSPeed:DISPlay:COLumn:ITEM<x>?
<x> = 1 to 6 (column number)
<Element> = {<NRf>|SIGMa|SIGMB|SIGMC} (<NRf> = 1 to 6)
Example :HSPEED:DISPLAY:COLUMN:ITEM1 1
:HSPEED:DISPLAY:COLUMN:ITEM1?
-> :HSPEED:DISPLAY:COLUMN:ITEM1 1

:HSPeed:DISPlay:COLumn:NUMber

Sets or queries the number of display columns of high speed data capturing mode.

Syntax :HSPeed:DISPlay:COLumn:NUMber {<NRf>}
:HSPeed:DISPlay:COLumn:NUMber?
<NRf> = 4 or 6
Example :HSPEED:DISPLAY:COLUMN:NUMBER 4
:HSPEED:DISPLAY:COLUMN:NUMBER?
-> :HSPEED:DISPLAY:COLUMN:NUMBER 4

:HSPeed:DISPlay:COLumn:RESet

Resets the column display items of high speed data capturing mode to their default values.

Syntax :HSPeed:DISPlay:COLumn:RESet
Example :HSPEED:DISPLAY:COLUMN:RESET

:HSPeed:DISPlay:FRAMe

Sets or queries the on/off status of the display’s data section frame in high speed data capturing mode.

Syntax :HSPeed:DISPlay:FRAMe {<Boolean>}
:HSPeed:DISPlay:FRAMe?
Example :HSPEED:DISPLAY:FRAME ON
:HSPEED:DISPLAY:FRAME?
-> :HSPEED:DISPLAY:FRAME 1
Description This command performs the same setting as the “:DISPlay:NUMeric:FRAMe” command.

:HSPeed:DISPlay:PAGE

Sets or queries the display page of high speed data capturing mode.

Syntax :HSPeed:DISPlay:PAGE {<NRf>}
:HSPeed:DISPlay:PAGE?
<NRf> = 1 or 2 (page number)
<NRf> = 1 to 4; on models with the motor evaluation function (/MTR) option or auxiliary input (/AUX) option
Example :HSPEED:DISPLAY:PAGE 1
:HSPEED:DISPLAY:PAGE?
-> :HSPEED:DISPLAY:PAGE 1

:HSPeed:DISPlay:POVer

Sets or queries the on/off status of the display of peak over-range information in high speed data capturing mode.

Syntax :HSPeed:DISPlay:POVer {<Boolean>}
:HSPeed:DISPlay:POVer?
Example :HSPEED:DISPLAY:POVER OFF
:HSPEED:DISPLAY:POVER?
-> :HSPEED:DISPLAY:POVER 0

:HSPeed:EXTSync

Sets or queries the on/off status of the high speed data capturing's external synchronization signal.

Syntax :HSPeed:EXTSync {<Boolean>}
:HSPeed:EXTSync?

Example :HSPEED:EXTSYNC OFF
:HSPEED:EXTSYNC ?
-> :HSPEED:EXTSYNC 0

:HSPeed:FILTer?

Queries all high speed data capturing filter settings.

Syntax :HSPeed:FILTer?

:HSPeed:FILTer[:HS]

Sets or queries the high speed data capturing digital filter (HS filter).

Syntax :HSPeed:FILTer[:HS] {OFF|<Frequency>}
:HSPeed:FILTer:HS?
OFF = Digital filter off
<Frequency> = 1 Hz to 1000 Hz (when the digital filter is on; cutoff frequency)

Example :HSPEED:FILTER:HS 100HZ
:HSPEED:FILTER:HS?
-> :HSPEED:FILTER:HS 100.0E+00

:HSPeed:FILTer:LINE?

Queries all high speed data capturing line filter settings.

Syntax :HSPeed:FILTer:LINE?

:HSPeed:FILTer:LINE[:ALL]

Sets the line filters of all the high speed data capturing elements.

Syntax :HSPeed:FILTer:LINE
[:ALL] {<Frequency>}
<Frequency> = 0.1 kHz to 100.0 kHz, 300 kHz (cutoff frequency)

Example :HSPEED:FILTER:LINE:ALL 300KHZ

Description
- Line filters are always on in high speed data capturing mode.
- This command sets the cutoff frequency. You can set the frequency between 0.1 kHz and 100.0 kHz with a resolution of 0.1 kHz.

:HSPeed:FILTer:LINE:ELEMent<x>

Sets or queries the line filter of the specified high speed data capturing element.

Syntax :HSPeed:FILTer:LINE:
ELEMent<x> {<Frequency>}
:HSPeed:FILTer:LINE:ELEMent<x>?
<x> = 1 to 6 (element)
<Frequency> = 0.1 kHz to 100.0 kHz, 300 kHz (cutoff frequency)

Example :HSPEED:FILTER:LINE:ELEMENT1 300KHZ
:HSPEED:FILTER:LINE:ELEMENT1?
-> :HSPEED:FILTER:LINE:
ELEMENT1 300.000E+03

Description
- Line filters are always on in high speed data capturing mode.
- This command sets the cutoff frequency. You can set the frequency between 0.1 kHz and 100.0 kHz with a resolution of 0.1 kHz.

:HSPeed:MAXCount?

Sets or queries the maximum number of data captures.

Syntax :HSPeed:MAXCount?

Example :HSPEED:MAXCOUNT? -> 506811

:HSPeed:MEASuring?

Queries all high speed data capturing voltage mode or current mode settings.

Syntax :HSPeed:MEASuring?

:HSPeed:MEASuring[:ALL]

Sets all voltage and current modes at the same time.

Syntax :HSPeed:MEASuring[:ALL] {RMS|MEAN|DC|RMEAN}

Example :HSPEED:MEASURING:ALL RMS

:HSPeed:MEASuring:{U<x>|I<x>}

Sets or queries the specified voltage or current mode.

Syntax :HSPeed:MEASuring:{U<x>|I<x>} {RMS|MEAN|DC|RMEAN}
:HSPeed:MEASuring:{U<x>|I<x>}?
<x> = 1 to 6 (element)

Example :HSPEED:MEASURING:U1 RMS
:HSPEED:MEASURING:U1?
-> :HSPEED:MEASURING:U1 RMS

:HSPeed:MEASuring:{UALL|IALL}

Sets all voltage or current modes at the same time.

Syntax :HSPeed:MEASuring:{UALL|IALL} {RMS|MEAN|DC|RMEAN}

Example :HSPEED:MEASURING:UALL RMS

:HSPeed:POVer?

Queries the high speed data capturing peak over-range information.

Syntax :HSPeed:POVer?

Example :HSPEED:POVER? -> 0

Description
- Returns the peak over-range information (the "Peak Over Status" that is displayed in the middle of the screen) during capturing.
- The peak over-range information of each element is mapped as shown below. For the response, the sum of the values of each bit is returned in decimal format. For example, a response of 16 indicates that a peak over-range is occurring at U3.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
--	--	Tq	Sp	I6	U6	I5	U5	I4	U4	I3	U3	I2	U2	I1	U1

Sp: Rotating speed or AUX1
Tq: Torque or AUX2

:HSPeed:RECord?

Queries all high speed data capturing settings for saving data to files.

Syntax :HSPeed:RECord?

:HSPeed:RECord:FILE?

Queries all settings related to the saving of acquired data to files.

Syntax :HSPeed:RECord:FILE?

Description The ":HSPeed:RECord:FILE:. . ." commands perform the same settings and queries as the ":STORe:FILE:. . ." commands. (Excluding the ":STORe:FILE:CONVert:. . ." commands.)

:HSPeed:RECord:FILE:ANAMing

Sets or queries the auto naming feature for saving acquired numeric data to files.

Syntax :HSPeed:RECord:FILE:ANAMing {OFF|NUMBering|DATE}
:HSPeed:RECord:FILE:ANAMing?

Example :HSPEED:RECORD:FILE
:ANAMING NUMBERING
:HSPEED:RECORD:FILE:ANAMING?
-> :HSPEED:RECORD:FILE
:ANAMING NUMBERING

:HSPeed:RECord:FILE:CDIRectory

Changes the directory that acquired numeric data will be saved to.

Syntax :HSPeed:RECord:FILE:CDIRectory {<String>}
<String> = Directory name

Example :HSPEED:RECORD:FILE:CDIRECTORY "RECORD"

Description Specify ".." to move up to the parent directory.

:HSPeed:RECord:FILE:CONVert?

Queries all settings related to the conversion of files of acquired numeric data into CSV format.

Syntax :HSPeed:RECord:FILE:CONVert?

:HSPeed:RECord:FILE:CONVert:ABORt

Aborts the conversion of the specified file of acquired numeric data to CSV format.

Syntax :HSPeed:RECord:FILE:CONVert:ABORt

Example :HSPEED:RECORD:FILE:CONVERT:ABORT

:HSPeed:RECord:FILE:CONVert:AUTO

Sets or queries the on/off status of the automatic conversion of files of acquired numeric data to CSV format.

Syntax :HSPeed:RECord:FILE:CONVert:AUTO {<Boolean>}
:HSPeed:RECord:FILE:CONVert:AUTO?

Example :HSPEED:RECORD:FILE:CONVERT:AUTO ON
:HSPEED:RECORD:FILE:CONVERT:AUTO?
-> :HSPEED:RECORD:FILE:CONVERT:AUTO 1

HSPeed:RECord:FILE:CONVert:CDIRectory

Changes the directory that acquired numeric data (CSV file) will be saved to.

Syntax :HSPeed:RECord:FILE:CONVert:CDIRectory {<String>}
<String> = Directory name

Example :HSPEED:RECORD:FILE:CONVERT:CDIRECTORY "RECORD"

Description Specify ".." to move up to the parent directory.

:HSPeed:RECord:FILE:CONVert:DRIVe

Sets the drive that acquired numeric data (CSV file) is saved to.

Syntax :HSPeed:RECord:FILE:CONVert:DRIVe {USER|RAM|USB[,<NRf>]|NETWork}
USER = Internal memory drive
RAM = Internal RAM disk
USB = USB memory device drive,
<NRf> = 0 or 1 (drive number)
NETWork = Network drive

Example :HSPEED:RECORD:FILE:CONVERT:DRIVE USER

:HSPeed:RECord:FILE:CONVert:EXECute

Converts the specified file of acquired numeric data to CSV format.

Syntax :HSPeed:RECord:FILE:CONVert:EXECute {<String>}
<String> = File name

Example :HSPEED:RECORD:FILE:CONVERT:EXECUTE "RECORD1"

Description
- Specify the file name without an extension.
- This command is an overlap command.

:HSPeed:RECord:FILE:CONVert:FREE?

Queries the free space (in bytes) on the drive that the acquired numeric data (CSV file) will be saved to.

Syntax :HSPeed:RECord:FILE:CONVert:FREE?

Example :HSPEED:RECORD:FILE:CONVERT:FREE?
-> 20912128

:HSPeed:RECord:FILE:CONVert:PATH?

Queries the absolute path of the directory that the acquired numeric data (CSV file) will be saved to.

Syntax :HSPeed:RECord:FILE:CONVert:PATH?

Example :HSPEED:RECORD:FILE:CONVERT:PATH?
-> "USB-0/RECORD"

:HSPeed:RECord:FILE:CONVert:SPMode

Sets or queries the on/off state of the save location path designation mode of the acquired numeric data (CSV path selection mode)

Syntax :HSPeed:RECord:FILE:CONVert:SPMode {<Boolean>}
:HSPeed:RECord:FILE:CONVert:SPMode?

Example :HSPEED:RECORD:FILE:CONVERT:SPMODE ON
:HSPEED:RECORD:FILE:CONVERT:SPMODE?
-> :HSPEED:RECORD:FILE:CONVERT:SPMODE 1

:HSPeed:RECord:FILE:DRIVe

Sets the drive that acquired numeric data is saved to.

Syntax :HSPeed:RECord:FILE:DRIVe {USER|RAM|USB[,<NRf>]|NETWork}
USER = Internal memory drive
RAM = Internal RAM disk
USB = USB memory device drive;
<NRf> = 0 or 1 (drive number)
NETWork = Network drive

Example :HSPEED:RECORD:FILE:DRIVE RAM

:HSPeed:RECord:FILE:FREE?

Queries the free space (in bytes) on the drive that the acquired numeric data will be saved to.

Syntax :HSPeed:RECord:FILE:FREE?

Example :HSPEED:RECORD:FILE:FREE? -> 20912128

:HSPeed:RECord:FILE:NAME

Sets or queries the name of the file that acquired numeric data will be saved to.

Syntax :HSPeed:RECord:FILE:NAME {<String>}
:HSPeed:RECord:FILE:NAME?
<String> = File name

Example :HSPEED:RECORD:FILE:NAME "RECORD1"
:HSPEED:RECORD:FILE:NAME?
-> :HSPEED:RECORD:FILE:NAME "RECORD1"

:HSPeed:RECord:FILE:PATH?

Queries the absolute path of the directory that the acquired numeric data will be saved to.

Syntax :HSPeed:RECord:FILE:PATH?

Example :HSPEED:RECORD:FILE:PATH?
-> "USB-0/RECORD"

:HSPeed:RECord:FILE:STATe?

Queries the status of the file save operation being performed on the acquired numeric data.

Syntax :HSPeed:RECord:FILE:STATe?

Example :HSPEED:RECORD:FILE:STATE? -> READY

Description The response is as follows:
READy = The file is closed (data capturing is in standby or the data has been captured and saved to the file).
RECord = The file is open (data is being captured).
STOP = The file is closed (data is being captured, but the file save operation has been stopped because an error occurred).
CONVert = Stored data is being converted to CSV format.
OFF = This instrument is not configured to save captured data to files, or this instrument is not in high speed data capturing mode.

:HSPeed:RECord:ITEM?

Queries all settings for the numeric data items that will be saved to a file.

Syntax :HSPeed:RECord:ITEM?

:HSPeed:RECord:ITEM:AUX<x>

Sets or queries whether numeric data (auxiliary input) is saved to a file.

Syntax :HSPeed:RECord:ITEM:AUX<x> {<Boolean>}
:HSPeed:RECord:ITEM:AUX<x>?
<x> = 1 or 2 (AUX input channel)

Example :HSPEED:RECORD:ITEM:AUX1 ON
:HSPEED:RECORD:ITEM:AUX1?
-> :HSPEED:RECORD:ITEM:AUX1 1

Description This is only valid on models with the auxiliary input (/AUX) option.

:HSPeed:RECord:ITEM:{I<x>|IA|IB|IC}

Sets or queries whether the specified element or wiring unit (ΣA, ΣB, or ΣC) of the numeric data (current) will be saved.

Syntax
```
:HSPeed:RECord:ITEM:{I<x>|IA|IB|
IC} {<Boolean>}
:HSPeed:RECord:ITEM:{I<x>|IA|IB|IC}?
```
<x> = 1 to 6 (element)
IA, IB, IC = IΣA, IΣB, IΣC

Example
```
:HSPEED:RECORD:ITEM:I1 ON
:HSPEED:RECORD:ITEM:I1?
-> :HSPEED:RECORD:ITEM:I1 1
```

:HSPeed:RECord:ITEM:{P<x>|PA|PB|PC}

Sets or queries whether the specified element or wiring unit (ΣA, ΣB, or ΣC) of the numeric data (active power) will be saved.

Syntax
```
:HSPeed:RECord:ITEM:{P<x>|PA|PB|
PC} {<Boolean>}
:HSPeed:RECord:ITEM:{P<x>|PA|PB|PC}?
```
<x> = 1 to 6 (element)
PA, PB, PC = PΣA, PΣB, PΣC

Example
```
:HSPEED:RECORD:ITEM:P1 ON
:HSPEED:RECORD:ITEM:P1?
-> :HSPEED:RECORD:ITEM:P1 1
```

:HSPeed:RECord:ITEM:{SPEed|TORQue|PM}

Sets or queries whether the rotating speed, torque, or motor output of the numeric data (motor) will be saved.

Syntax
```
:HSPeed:RECord:ITEM:{SPEed|TORQue|
PM} {<Boolean>}
:HSPeed:RECord:ITEM:{SPEed|TORQue|
PM}?
```

Example
```
:HSPEED:RECORD:ITEM:SPEED ON
:HSPEED:RECORD:ITEM:SPEED?
-> :HSPEED:RECORD:ITEM:SPEED 1
```

Description This is only valid on models with the motor evaluation function (/MTR) option.

:HSPeed:RECord:ITEM:{U<x>|UA|UB|UC}

Sets or queries whether the specified element or wiring unit (ΣA, ΣB, or ΣC) of the numeric data (voltage) will be saved.

Syntax
```
:HSPeed:RECord:ITEM:{U<x>|UA|UB|
UC} {<Boolean>}
:HSPeed:RECord:ITEM:{U<x>|UA|UB|UC}?
```
<x> = 1 to 6 (element)
UA, UB, UC = UΣA, UΣB, UΣC

Example
```
:HSPEED:RECORD:ITEM:U1 ON
:HSPEED:RECORD:ITEM:U1?
-> :HSPEED:RECORD:ITEM:U1 1
```

:HSPeed:RECord:ITEM:PRESet:ALL

Sets, at the same time, whether all numeric data items will be saved.

Syntax
```
:HSPeed:RECord:ITEM:PRESet:
ALL {<Boolean>}
```

Example
```
:HSPEED:RECORD:ITEM:PRESET:ALL ON
```

:HSPeed:RECord:ITEM:PRESet:{ELEMent<x>|SIGMA|SIGMB|SIGMC}

Sets, at the same time, whether the specified element or wiring unit (ΣA, ΣB, or ΣC) of all the types of numeric data will be saved.

Syntax
```
:HSPeed:RECord:ITEM:PRESet:
{ELEMent<x>|SIGMA|SIGMB|
SIGMC} {<Boolean>}
```
<x> = 1 to 6

Example
```
:HSPEED:RECORD:ITEM:PRESET:
ELEMENT1 ON
```

:HSPeed:RECord:ITEM:PRESet:{U|I|P|MOTor|AUX}

Sets, at the same time, whether the specified functions of all types of numeric data will be saved.

Syntax
```
:HSPeed:RECord:ITEM:PRESet:{U|I|P|
MOTor|AUX} {<Boolean>}
```

Example
```
:HSPEED:RECORD:ITEM:PRESET:U ON
```

Description
- MOTor is only valid on models with the motor evaluation function (/MTR) option.
- AUX is only valid on models with the auxiliary input (/AUX) option.

:HSPeed:RECord[:STATe]

Sets or queries whether acquired numeric data is saved to a file.

Syntax
```
:HSPeed:RECord[:STATe] {<Boolean>}
:HSPeed:RECord:STATe?
```

Example
```
:HSPEED:RECORD:STATE ON
:HSPEED:RECORD:STATE?
-> :HSPEED:RECORD:STATE 1
```

:HSPeed:STARt

Starts data capturing.

Syntax
```
:HSPeed:STARt
```

Example
```
:HSPEED:START
```

:HSPeed:STATe?

Queries the status of high speed data capturing.

Syntax :HSPeed:STATe?

Example ::HSPEED:STATE? -> READY

Description The response is as follows:
INIT = High-speed data capturing is being initialized.
READy = High-speed data capturing is in stand by.
STARt = High-speed data capturing is being performed.
OFF = This instrument is not in high speed data capturing mode.

:HSPeed:STOP

Stops data capturing.

Syntax :HSPeed:STOP

Example :HSPEED:STOP

:HSPeed:TRIGger?

Queries all high speed data capturing trigger settings.

Syntax :HSPeed:TRIGger?

:HSPeed:TRIGger:LEVel

Sets or queries the trigger level.

Syntax :HSPeed:TRIGger:LEVel {<NRf>}
:HSPeed:TRIGger:LEVel?
<NRf> = ?100.0 to 100.0(%)

Example :HSPEED:TRIGGER:LEVEL 0
:HSPEED:TRIGGER:LEVEL?
-> :HSPEED:TRIGGER:LEVEL 0.0

Description This command performs the same setting as the ":DISPlay:WAVE:TRIGger:LEVel" command.

:HSPeed:TRIGger:MODE

Sets or queries the trigger mode.

Syntax :HSPeed:TRIGger:MODE {AUTO|NORMal|OFF}
:HSPeed:TRIGger:MODE?

Example :HSPEED:TRIGGER:MODE AUTO
:HSPEED:TRIGGER:MODE?
-> :HSPEED:TRIGGER:MODE AUTO

Description This command performs the same setting as the ":DISPlay:WAVE:TRIGger:MODE" command.

:HSPeed:TRIGger:SLOPe

Sets or queries the trigger slope.

Syntax :HSPeed:TRIGger:SLOPe {RISE|FALL|BOTH}
:HSPeed:TRIGger:SLOPe?

Example :HSPEED:TRIGGER:SLOPE RISE
:HSPEED:TRIGGER:SLOPE?
-> :HSPEED:TRIGGER:SLOPE RISE

Description This command performs the same setting as the ":DISPlay:WAVE:TRIGger:SLOPe" command.

:HSPeed:TRIGger:SOURce

Sets or queries the trigger source.

Syntax :HSPeed:TRIGger:SOURce {U<x>|I<x>|EXTernal}
:HSPeed:TRIGger:SOURce?
<x> = 1 to 6 (element)
EXTernal = External trigger input (Ext Clk)

Example :HSPEED:TRIGGER:SOURCE U1
:HSPEED:TRIGGER:SOURCE?
-> :HSPEED:TRIGGER:SOURCE U1

Description This command performs the same setting as the ":DISPlay:WAVE:TRIGger:SOURce" command.

5.12 IMAGe Group

The commands in this group deal with saving screen image data. You can perform the same operations and make the same settings and queries that you can by pressing IMAGE SAVE and MENU (SHIFT+IMAGE SAVE) on the front panel.

:IMAGe?

Queries all screen image data output settings.

Syntax :IMAGe?

:IMAGe:ABORt

Aborts a screen image data output operation.

Syntax :IMAGe:ABORt
Example :IMAGE:ABORT

:IMAGe:COLor

Sets or queries the color tone of the screen image data that will be saved.

Syntax :IMAGe:COLor {OFF|COLor|REVerse|GRAY}
:IMAGe:COLor?
Example :IMAGE:COLOR OFF
:IMAGE:COLOR?
-> :IMAGE:COLOR OFF

:IMAGe:COMMent

Sets or queries the comment displayed at the bottom of the screen.

Syntax :IMAGe:COMMent {<String>}
:IMAGe:COMMent?
<String> = Up to 30 characters
Example :IMAGE:COMMENT "THIS IS TEST."
:IMAGE:COMMENT?
-> :IMAGE:COMMENT "THIS IS TEST."

:IMAGe:EXECute

Executes a screen image data output operation.

Syntax :IMAGe:EXECute
Example :IMAGE:EXECUTE

:IMAGe:FORMat

Sets or queries the format of the screen image data that will be saved.

Syntax :IMAGe:FORMat {BMP|PNG|JPEG}
:IMAGe:FORMat?
Example :IMAGE:FORMAT BMP
:IMAGE:FORMAT?
-> :IMAGE:FORMAT BMP

:IMAGe:SAVE?

Queries all screen image data save settings.

Syntax :IMAGe:SAVE?

:IMAGe:SAVE:ANAMing

Sets or queries the auto naming feature for saving files.

Syntax :IMAGe:SAVE:ANAMing {OFF|NUMBering|DATE}
:IMAGe:SAVE:ANAMing?
Example :IMAGE:SAVE:ANAMING NUMBERING
:IMAGE:SAVE:ANAMING?
-> :IMAGE:SAVE:ANAMING NUMBERING

:IMAGe:SAVE:CDIRectory

Changes the directory that screen image data is saved to.

Syntax :IMAGe:SAVE:CDIRectory {<String>}
<String> = Directory name
Example :IMAGE:SAVE:CDIRECTORY "IMAGE"
Description Specify ".." to move up to the parent directory.

:IMAGe:SAVE:DRIVe

Sets the drive that screen image data is saved to.

Syntax :IMAGe:SAVE:DRIVe {USER|RAM|USB[,<NRf>]|NETWork}
USER = Internal memory drive
RAM = Internal RAM disk
USB = USB memory device drive,
<NRf> = 0 or 1 (drive number)
NETWork = Network drive
Example :IMAGE:SAVE:DRIVE USER

:IMAGe:SAVE:FREE?

Queries the free space (in bytes) on the drive that the screen image data is saved to.

Syntax :IMAGe:SAVE:FREE?
Example :IMAGE:SAVE:FREE? -> 20912128

:IMAGe:SAVE:NAME

Sets or queries the name of the file that will be saved.

Syntax `:IMAGe:SAVE:NAME {<String>}`
`:IMAGe:SAVE:NAME?`
<String> = File name

Example `:IMAGE:SAVE:NAME "IMAGE1"`
`:IMAGE:SAVE:NAME?`
`-> :IMAGE:SAVE:NAME "IMAGE1"`

Description
- Use the :IMAGe:SAVE:DRIVe command to set the save destination drive and the :IMAGe:SAVE:CDIRectory command to set the directory.
- You can query the path that screen image data is saved to by using the :IMAGe:SAVE:PATH? command.
- Specify the file name without an extension.

:IMAGe:SAVE:PATH?

Queries the absolute path of the directory that the screen image data is saved to.

Syntax `:IMAGe:SAVE:PATH?`

Example `:IMAGE:SAVE:PATH? -> "USB-0/IMAGE"`

:IMAGe:SEND?

Queries the screen image data.

Syntax `:IMAGe:SEND?`

Example `:IMAGE:SEND?`
`-> #N (N-digit byte number)(data byte sequence)`

Description N, the number of digits in the data byte number, varies depending on the output data size.

5.13 INPut Group

The commands in this group deal with the measurement conditions of the input elements.
You can make the same settings and queries that you can make by pressing the keys in the measurement condition setup area (the area outlined in blue), SCALING, LINE FILTER, FREQ FILTER (SHIFT+LINE FILTER), SYNC SOURCE, NULL, and NULL SET (SHIFT+NULL) on the front panel.

:INPut?

Queries all input element settings.

Syntax `:INPut?`

[:INPut]:CFACtor

Sets or queries the crest factor.

Syntax `[:INPut]:CFACtor {<NRf>|A6}`
`[:INPut]:CFACtor?`
<NRf> = 3, 6
A6 = Display range expand mode (6A) for crest factor 6

Example
```
:INPUT:CFACTOR 3
:INPUT:CFACTOR?
-> :INPUT:CFACTOR 3
```

[:INPut]:CURRent?

Queries all electric current measurement settings.

Syntax `[:INPut]:CURRent?`

[:INPut]:CURRent:AUTO?

Queries the electric current auto range on/off statuses of all elements.

Syntax `[:INPut]:CURRent:AUTO?`

[:INPut]:CURRent:AUTO[:ALL]

Collectively sets the electric current auto range on/off status of all elements.

Syntax `[:INPut]:CURRent:AUTO[:ALL] {<Boolean>}`

Example `:INPUT:CURRENT:AUTO:ALL ON`

[:INPut]:CURRent:AUTO:ELEMent<x>

Sets or queries the electric current auto range on/off status of the specified element.

Syntax `[:INPut]:CURRent:AUTO:ELEMent<x> {<Boolean>}`
`[:INPut]:CURRent:AUTO:ELEMent<x>?`
<x> = 1 to 6 (element)

Example
```
:INPUT:CURRENT:AUTO:ELEMENT1 ON
:INPUT:CURRENT:AUTO:ELEMENT1?
-> :INPUT:CURRENT:AUTO:ELEMENT1 1
```

[:INPut]:CURRent:AUTO:{SIGMA|SIGMB|SIGMC}

Collectively sets the electric current auto range on/off status of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).

Syntax `[:INPut]:CURRent:AUTO:{SIGMA|SIGMB|SIGMC} {<Boolean>}`

Example `:INPUT:CURRENT:AUTO:SIGMA ON`

Description SIGMA, SIGMB, or SIGMC is invalid if the wiring system setting ([:INPut]:WIRing) is made in such a way that the corresponding wiring unit (ΣA, ΣB, or ΣC) does not exist.

[:INPut]:CURRent:CONFig?

Queries the valid electric current ranges of all elements.

Syntax `[:INPut]:CURRent:CONFig?`

[:INPut]:CURRent:CONFig[:ALL]

Collectively sets the valid electric current range of all elements.

Syntax `[:INPut]:CURRent:CONFig[:ALL] {ALL|<Current>[,<Current>][,<Current>]...}`
ALL = All ranges are valid.

50 A input elements

- When the crest factor is set to 3:
  <Current> = 1 A, 2 A, 5 A, 10 A, 20 A, 50 A
- When the crest factor is set to 6 or 6A:
  <Current> = 500 mA, 1 A, 2.5 A, 5 A, 10 A, 25 A

5 A input elements

- When the crest factor is set to 3:
  <Current> = 10 mA, 20 mA, 50 mA, 100 mA, 200 mA, 500 mA, 1 A, 2 A, 5 A
- When the crest factor is set to 6 or 6A:
  <Current> = 5 mA, 10 mA, 25 mA, 50 mA, 100 mA, 250 mA, 500 mA, 1 A, 2.5 A

Example
```
:INPUT:CURRENT:CONFIG:ALL ALL
:INPUT:CURRENT:CONFIG:ALL 50,10,5,1
```

Description In the parameters, list all the electric current ranges that you want to enable. To enable all the ranges, specify the parameter “ALL.”

[:INPut]:CURRent:CONFig:ELEMent<x>

Sets or queries the valid electric current range of the specified element.

Syntax
```
[:INPut]:CURRent:CONFig:
ELEMent<x> {ALL|<Current>[,<Current>]
[,<Current>]...}
[:INPut]:CURRent:CONFig:ELEMent<x>?
```
<x> = 1 to 6 (element)
ALL = All ranges are valid.
<Current> = See [:INPut]:CURRent:CONFig[:ALL]

Example
```
:INPUT:CURRENT:CONFIG:ELEMENT1 ALL
:INPUT:CURRENT:CONFIG:ELEMENT1?
-> :INPUT:CURRENT:CONFIG:ELEMENT1 ALL
:INPUT:CURRENT:CONFIG:
ELEMENT1 50,10,5,1
:INPUT:CURRENT:CONFIG:ELEMENT1?
-> :INPUT:CURRENT:CONFIG:
ELEMENT1 50.0E+00,10.0E+00,5.0E+00,
1.0E+00
```

Description In the parameters, list all the electric current ranges that you want to enable. To enable all the ranges, specify the parameter “ALL.”

[:INPut]:CURRent:EXTSensor?

Queries all external current sensor range settings.

Syntax
```
[:INPut]:CURRent:EXTSensor?
```

Description This command is only valid on models with the external current sensor input (/EX) option.

[:INPut]:CURRent:EXTSensor:CONFig?

Queries the valid external current sensor ranges of all elements.

Syntax
```
[:INPut]:CURRent:EXTSensor:CONFig?
```

[:INPut]:CURRent:EXTSensor:CONFig[:ALL]

Collectively sets the valid external current sensor range of all elements.

Syntax
```
[:INPut]:CURRent:EXTSensor:
CONFig[:ALL] {ALL|<Voltage>[,
<Voltage>][,<Voltage>]...}
```
ALL = All ranges are valid.
- When the crest factor is set to 3:
  <Voltage> = 50 mV, 100 mV, 200 mV, 500 mV, 1 V, 2 V, 5 V, 10 V
- When the crest factor is set to 6 or 6A:
  <Voltage> = 25 mV, 50 mV, 100 mV, 250 mV, 500 mV, 1 V, 2.5 V, 5 V

Example
```
:INPUT:CURRENT:EXTSENSOR:CONFIG:
ALL ALL
:INPUT:CURRENT:EXTSENSOR:CONFIG:
ALL 10,5,2,1
```

Description In the parameters, list all the external current sensor ranges that you want to enable. To enable all the ranges, specify the parameter “ALL.”

[:INPut]:CURRent:EXTSensor:CONFig:ELEMent<x>

Sets or queries the valid external current sensor ranges of the specified element.

Syntax
```
[:INPut]:CURRent:EXTSensor:CONFig:
ELEMent<x> {ALL|<Voltage>[,
<Voltage>][,<Voltage>]...}
[:INPut]:CURRent:EXTSensor:CONFig:
ELEMent<x>?
```
<x> = 1 to 6 (element)
ALL = All ranges are valid.
<Voltage> = See [:INPut]:CURRent:EXTSensor:CONFig[:ALL]

Example
```
:INPUT:CURRENT:EXTSENSOR:CONFIG:
ELEMENT1 ALL
:INPUT:CURRENT:EXTSENSOR:CONFIG:
ELEMENT1?
-> :INPUT:CURRENT:EXTSENSOR:CONFIG:
ELEMENT1 ALL
:INPUT:CURRENT:EXTSENSOR:CONFIG:
ELEMENT1 10,5,2,1
:INPUT:CURRENT:EXTSENSOR:CONFIG:
ELEMENT1?
-> :INPUT:CURRENT:EXTSENSOR:CONFIG:
ELEMENT1 10.00E+00,5.00E+00,2.00E+00,
1.00E+00
```

Description In the parameters, list all the external current sensor ranges that you want to enable. To enable all the ranges, specify the parameter “ALL.”

[:INPut]:CURRent:EXTSensor:DISPlay

Sets or queries the display mode of the external current sensor range.

Syntax
```
[:INPut]:CURRent:EXTSensor:
DISPlay {DIRect|MEASure}
[:INPut]:CURRent:EXTSensor:DISPlay?
```

Example
```
:INPUT:CURRENT:EXTSENSOR:
DISPLAY DIRECT
:INPUT:CURRENT:EXTSENSOR:DISPLAY?
-> :INPUT:CURRENT:EXTSENSOR:
DISPLAY DIRECT
```

[:INPut]:CURRent:EXTSensor:POJump?

Queries the jump destination ranges of all elements that are used when a current peak over-range occurs.

Syntax
```
[:INPut]:CURRent:EXTSensor:POJump?
```

[:INPut]:CURRent:EXTSensor:POJump[:ALL]

Collectively sets the jump destination range of all elements that is used when a current peak over-range occurs.

Syntax `[:INPut]:CURRent:EXTSensor:POJump[:ALL] {OFF|<Voltage>}`

OFF = No jump destination current range

- When the crest factor is set to 3:
  <Voltage> = 50 mV, 100 mV, 200 mV, 500 mV, 1 V, 2 V, 5 V, 10 V
- When the crest factor is set to 6 or 6A:
  <Voltage> = 25 mV, 50 mV, 100 mV, 250 mV, 500 mV, 1 V, 2.5 V, 5 V

Example
```
:INPUT:CURRENT:EXTSENSOR:POJUMP:ALL OFF
```

[:INPut]:CURRent:EXTSensor:POJump:ELEMent<x>

Sets or queries the jump destination range of the specified element that is used when a current peak over-range occurs.

Syntax
```
[:INPut]:CURRent:EXTSensor:POJump:ELEMent<x> {OFF|<Voltage>}
[:INPut]:CURRent:EXTSensor:POJump:ELEMent<x>?
```
<x> = 1 to 6 (element)

OFF = No jump destination current range

<Voltage> = See [:INPut]:CURRent:EXTSensor:POJump[:ALL]

Example
```
:INPUT:CURRENT:EXTSENSOR:POJUMP:ELEMENT1 10V
:INPUT:CURRENT:EXTSENSOR:POJUMP:ELEMENT1?
-> :INPUT:CURRENT:EXTSENSOR:POJUMP:ELEMENT1 10.00E+00
```

[:INPut]:CURRent:POJump?

Queries the jump destination ranges of all elements that are used when a current peak over-range occurs.

Syntax `[:INPut]:CURRent:POJump?`

[:INPut]:CURRent:POJump[:ALL]

Collectively sets the jump destination range of all elements that is used when a current peak over-range occurs.

Syntax `[:INPut]:CURRent:POJump[:ALL] {OFF|<Current>}`

OFF = No jump destination current range

50 A input elements

- When the crest factor is set to 3:
  <Current> = 1 A, 2 A, 5 A, 10 A, 20 A, 50 A
- When the crest factor is set to 6 or 6A:
  <Current> = 500 mA, 1 A, 2.5 A, 5 A, 10 A, 25 A

5 A input elements

- When the crest factor is set to 3:
  <Current> = 10 mA, 20 mA, 50 mA, 100 mA, 200 mA, 500 mA, 1 A, 2 A, 5 A
- When the crest factor is set to 6 or 6A:
  <Current> = 5 mA, 10 mA, 25 mA, 50 mA, 100 mA, 250 mA, 500 mA, 1 A, 2.5 A

Example
```
:INPUT:CURRENT:POJUMP:ALL OFF
```

[:INPut]:CURRent:POJump:ELEMent<x>

Sets or queries the jump destination range of the specified element that is used when a current peak over-range occurs.

Syntax
```
[:INPut]:CURRent:POJump:ELEMent<x> {OFF|<Current>}
[:INPut]:CURRent:POJump:ELEMent<x>?
```
<x> = 1 to 6 (element)

OFF = No jump destination current range

<Current> = See [:INPut]:CURRent:POJump[:ALL]

Example
```
:INPUT:CURRENT:POJUMP:ELEMENT1 50A
:INPUT:CURRENT:POJUMP:ELEMENT1?
-> :INPUT:CURRENT:POJUMP:ELEMENT1 50.0E+00
```

[:INPut]:CURRent:RANGe?

Queries the electric current ranges of all elements.

Syntax `[:INPut]:CURRent:RANGe?`

[:INPut]:CURRent:RANGe[:ALL]

Collectively sets the electric current range of all elements.

Syntax `[:INPut]:CURRent:RANGe[:ALL] {<Current>|(EXTernal,<Voltage>)}`

50 A input elements

- When the crest factor is set to 3:
  <Current> = 1 A, 2 A, 5 A, 10 A, 20 A, 50 A (for direct current input)
  <Voltage> = 50 mV, 100 mV, 200 mV, 500 mV, 1 V, 2 V, 5 V, 10 V (for external current sensor input)
- When the crest factor is set to 6 or 6A:
  <Current> = 500 mA, 1 A, 2.5 A, 5 A, 10 A, 25 A (for direct current input)
  <Voltage> = 25 mV, 50 mV, 100 mV, 250 mV, 500 mV, 1 V, 2.5 V, 5 V (for external current sensor input)

5 A input elements

- When the crest factor is set to 3:
  <Current> = 10 mA, 20 mA, 50 mA, 100 mA, 200 mA, 500 mA, 1 A, 2 A, 5 A (for direct current input)
  <Voltage> = 50 mV, 100 mV, 200 mV, 500 mV, 1 V, 2 V, 5 V, 10 V (for external current sensor input)
- When the crest factor is set to 6 or 6A:
  <Current> = 5 mA, 10 mA, 25 mA, 50 mA, 100 mA, 250 mA, 500 mA, 1 A, 2.5 A (for direct current input)
  <Voltage> = 25 mV, 50 mV, 100 mV, 250 mV, 500 mV, 1 V, 2.5 V, 5 V (for external current sensor input)

Example
```
:INPUT:CURRENT:RANGE:ALL 50A
:INPUT:CURRENT:RANGE:ALL EXTERNAL,10V
```

Description EXTernal and <Voltage> can only be selected on models with the external current sensor input (/EX) option.

[:INPut]:CURRent:RANGe:ELEMent<x>

Sets or queries the electric current range of the specified element.

Syntax
```
[:INPut]:CURRent:RANGe:ELEMent<x> {<Current>|(EXTernal,<Voltage>)}
[:INPut]:CURRent:RANGe:ELEMent<x>?
```
<x> = 1 to 6 (element)
<Current>,<Voltage> = See [:INPut]:CURRent:RANGe[:ALL]

Example
```
:INPUT:CURRENT:RANGE:ELEMENT1 50A
:INPUT:CURRENT:RANGE:ELEMENT1?
-> :INPUT:CURRENT:RANGE:ELEMENT1 50.0E+00
:INPUT:CURRENT:RANGE:ELEMENT1 EXTERNAL,10V
:INPUT:CURRENT:RANGE:ELEMENT1?
-> :INPUT:CURRENT:RANGE:ELEMENT1 EXTERNAL,10.00E+00
```

[:INPut]:CURRent:RANGe:{SIGMA|SIGMB|SIGMC}

Collectively sets the electric current range of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).

Syntax
```
[:INPut]:CURRent:RANGe:{SIGMA|SIGMB|SIGMC} {<Current>|(EXTernal,<Voltage>)}
```
<Current>,<Voltage> = See [:INPut]:CURRent:RANGe[:ALL]

Example
```
:INPUT:CURRENT:RANGE:SIGMA 50A
:INPUT:CURRENT:RANGE:SIGMA EXTERNAL,10V
```

Description SIGMA, SIGMB, or SIGMC is invalid if the wiring system setting ([:INPut]:WIRing) is made in such a way that the corresponding wiring unit (ΣA, ΣB, or ΣC) does not exist.

[:INPut]:CURRent:SPReset?

Queries the external current sensor conversion ratio presets of all elements.

Syntax `[:INPut]:CURRent:SPReset?`

Description This command is only valid on models with the external current sensor input (/EX) option.

[:INPut]:CURRent:SPReset[:ALL]

Collectively sets the external current sensor conversion ratio presets of all elements.

Syntax `[:INPut]:CURRent:SPReset[:ALL] {OTHERS|SHUNT20|SHUNT10|SHUNT5|CT1000S}`

Example `:INPUT:CURRENT:SPRESET:ALL SHUNT20`

[:INPut]:CURRent:SPReset:ELEMent<x>

Sets or queries the external current sensor conversion ratio preset of the specified element.

Syntax
```
[:INPut]:CURRent:SPReset:ELEMent<x> {OTHERS|SHUNT20|SHUNT10|SHUNT5|CT1000S}
[:INPut]:CURRent:SPReset:ELEMent<x>?
```
<x> = 1 to 6 (element)

Example
```
:INPUT:CURRENT:SPRESET:ELEMENT1 SHUNT20
:INPUT:CURRENT:SPRESET:ELEMENT1?
-> :INPUT:CURRENT:SPRESET:ELEMENT1 SHUNT20
```

Description If a preset is queried when the preset value and the external current sensor conversion ratio value are different, "A" is added to the preset name.

[:INPut]:CURRent:SPReset:{SIGMA|SIGMB|SIGMC}

Collectively sets the external current sensor conversion ratio presets of all the elements that belong to the specified wiring unit (ΣA, ΣB or ΣC).

Syntax `[:INPut]:CURRent:SPReset:{SIGMA|SIGMB|SIGMC} {OTHERS|SHUNT20|SHUNT10|SHUNT5|CT1000S}`

Example `:INPUT:CURRENT:SPRESET:SIGMA SHUNT20`

Description SIGMA, SIGMB, or SIGMC is invalid if the wiring system setting ([:INPut]:WIRing) is made in such a way that the corresponding wiring unit (ΣA, ΣB, or ΣC) does not exist.

[:INPut]:CURRent:SRATio?

Queries the external current sensor conversion ratios of all elements.

Syntax `[:INPut]:CURRent:SRATio?`

Description This command is only valid on models with the external current sensor input (/EX) option.

[:INPut]:CURRent:SRATio[:ALL]

Collectively sets the external current sensor conversion ratios of all elements.

Syntax `[:INPut]:CURRent:SRATio[:ALL] {<NRf>}`
<NRf> = 0.0001 to 99999.9999

Example `:INPUT:CURRENT:SRATIO:ALL 10`

[:INPut]:CURRent:SRATio:ELEMent<x>

Sets or queries the external current sensor conversion ratio of the specified element.

Syntax `[:INPut]:CURRent:SRATio:ELEMent<x> {<NRf>}`
`[:INPut]:CURRent:SRATio:ELEMent<x>?`
<x> = 1 to 6 (element)
<NRf> = 0.0001 to 99999.9999

Example
```
:INPUT:CURRENT:SRATIO:ELEMENT1 10
:INPUT:CURRENT:SRATIO:ELEMENT1?
-> :INPUT:CURRENT:SRATIO:
ELEMENT1 10.0000
```

[:INPut]:CURRent:SRATio:{SIGMA|SIGMB|SIGMC}

Collectively sets the external current sensor conversion ratios of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).

Syntax `[:INPut]:CURRent:SRATio:{SIGMA|SIGMB|SIGMC} {<NRf>}`
<NRf> = 0.0001 to 99999.9999

Example `:INPUT:CURRENT:SRATIO:SIGMA 10`

Description SIGMA, SIGMB, or SIGMC is invalid if the wiring system setting ([:INPut]:WIRing) is made in such a way that the corresponding wiring unit (ΣA, ΣB, or ΣC) does not exist.

[:INPut]:ESELect

Sets or queries the element whose measurement range will be set.

Syntax `[:INPut]:ESELect {<NRf>|ALL}`
`[:INPut]:ESELect?`
<NRf> = 1 to 6 (element)

Example
```
:INPUT:ESELECT 1
:INPUT:ESELECT? -> :INPUT:ESELECT 1
```

[:INPut]:FILTer?

Queries all input filter settings.

Syntax `[:INPut]:FILTer?`

[:INPut]:FILTer:FAUTo?

Queries the frequency filter (for when the data update interval is Auto) of all elements.

Syntax `[:INPut]:FILTer:FAUTO?`

[:INPut]:FILTer:FAUTo[:ALL]

Collectively sets the frequency filter (for when the data update interval is Auto) of all elements.

Syntax `[:INPut]:FILTer:FAUTo[:ALL] {OFF|<Frequency>}`
OFF = Frequency filter (for when the data update interval is Auto) off
<Frequency> = 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 12.8, 25.6 (kHz) (when the frequency filter (for when the data update interval is Auto) is on; cutoff frequency)

Example `:INPUT:FILTER:FAUTO:ALL OFF`

Description This command is valid only when the data update interval is Auto.

[:INPut]:FILTer:FAUTo:ELEMent<x>

Sets or queries the frequency filter (for when the data update interval is Auto) of the specified element.

Syntax `[:INPut]:FILTer:FAUTo:ELEMent<x> {OFF|<Frequency>}`
`[:INPut]:FILTer:FAUTo:ELEMent<x>?`
<x> = 1 to 6 (element)
OFF = Frequency filter (for when the data update interval is Auto) off
<Frequency> = 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 12.8, 25.6 (kHz) (when the frequency filter (for when the data update interval is Auto) is on; cutoff frequency)

Example
```
:INPUT:FILTER:FAUTO:ELEMNT1?
-> :INPUT:FILTER:FAUTO:
ELEMNT1 100.0E+00
```

Description This command is valid only when the data update interval is Auto.

[:INPut]:FILTer:FREQuency?

Queries the frequency filters of all elements.

Syntax `[:INPut]:FILTer:FREQuency?`

[:INPut]:FILTer:FREQuency[:ALL]

Collectively sets the frequency filter of all elements.

Syntax `[:INPut]:FILTer:FREQuency[:ALL] {OFF|<Frequency>}`
OFF = Frequency filter off
<Frequency> = 100 Hz, 1 kHz (when the line filter is on; cutoff frequency)

Example `:INPUT:FILTER:FREQUENCY:ALL OFF`

[:INPut]:FILTer:FREQuency:ELEMent<x>

Sets or queries the frequency filter of the specified element.

Syntax `[:INPut]:FILTer:FREQuency:ELEMent<x> {OFF|<Frequency>}`
`[:INPut]:FILTer:FREQuency:ELEMent<x>?`
<x> = 1 to 6 (element)
OFF = Frequency filter off
<Frequency> = 100 Hz, 1 kHz (when the line filter is on; cutoff frequency)

Example
```
:INPUT:FILTER:FREQUENCY:ELEMENT1 100HZ
:INPUT:FILTER:FREQUENCY:ELEMENT1?
-> :INPUT:FILTER:FREQUENCY:ELEMENT1 100.0E+00
```

[:INPut]:FILTer:LINE?

Queries the line filters of all elements.

Syntax `[:INPut]:FILTer:LINE?`

[:INPut]:FILTer[:LINE][:ALL]

Collectively sets the line filter of all elements.

Syntax `[:INPut]:FILTer[:LINE][:ALL] {OFF|<Frequency>}`
OFF = Line filter off
<Frequency> = 0.1 kHz to 100.0 kHz, 300 kHz, 1 MHz (when the line filter is on; cutoff frequency)

Example `:INPUT:FILTER:LINE:ALL OFF`

Description You can set the frequency between 0.1 kHz and 100.0 kHz with a resolution of 0.1 kHz.

[:INPut]:FILTer[:LINE]:ELEMent<x>

Sets or queries the line filter of the specified element.

Syntax `[:INPut]:FILTer[:LINE]:ELEMent<x> {OFF|<Frequency>}`
`[:INPut]:FILTer[:LINE]:ELEMent<x>?`
<x> = 1 to 6 (element)
OFF = Line filter off
<Frequency> = 0.1 kHz to 100.0 kHz, 300 kHz, 1 MHz (when the line filter is on; cutoff frequency)

Example
```
:INPUT:FILTER:LINE:ELEMENT1 0.5KHZ
:INPUT:FILTER:LINE:ELEMENT1?
-> :INPUT:FILTER:LINE:ELEMENT1 500.0E+00
```

Description You can set the frequency between 0.1 kHz and 100.0 kHz with a resolution of 0.1 kHz.

[:INPut]:FILTer[:LINE]:{SIGMA|SIGMB|SIGMC}

Collectively sets the line filter of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).

Syntax `[:INPut]:FILTer[:LINE]:{SIGMA|SIGMB|SIGMC} {OFF|<Frequency>}`
OFF = Line filter off
<Frequency> = 0.1 kHz to 100.0 kHz, 300 kHz, 1 MHz (when the line filter is on; cutoff frequency)

Example `:INPUT:FILTER:LINE:SIGMA 300KHZ`

Description You can set the frequency between 0.1 kHz and 100.0 kHz with a resolution of 0.1 kHz.

[:INPut]:INDependent

Sets or queries the on/off status of independent input element configuration.

Syntax `[:INPut]:INDependent {<Boolean>}`
`[:INPut]:INDependent?`

Example
```
:INPUT:INDEPENDENT OFF
:INPUT:INDEPENDENT?
-> :INPUT:INDEPENDENT 0
```

Description This command is only valid on models with 2 to 6 elements.

[:INPut]:MODUle?

Queries the input element type.

Syntax `[:INPut]:MODUle? {<NRf>}`
`[:INPut]:MODUle?`
<NRf> = 1 to 6 (element)

Example `:INPUT:MODULE? 1 -> 50`
`:INPUT:MODULE? -> 50,50,50,50,50,50`

Description
- The response is as follows:
  50 = 50 A input element (maximum current range = 50 A)
  5 = 5 A input element (maximum current range = 5 A)
  0 = No input element
- If the parameter is omitted, the input element types of all elements are output in order, starting with element 1.

[:INPut]:NULL:CONDition:{SPEed|TORQue|AUX<x>}

Queries the status of the NULL operation of rotating speed, torque, or AUX.

Syntax `[:INPut]:NULL:CONDition:{SPEed|TORQue|AUX<x>}?`
<x> = 1 or 2 (AUX input channel)

Example `:INPUT:NULL:CONDITION:SPEED? -> 1`

Description
- The response is as follows:
  0 = NULL feature off
  1 = NULL feature in operation
- SPEed and TORQue are only valid on models with the motor evaluation function (/MTR) option.
- AUX<x> is only valid on models with the auxiliary input (/AUX) option.

[:INPut]:NULL:CONDition:{U<x>|I<x>}

Queries the status of the voltage or current NULL operation of the specified element.

Syntax `[:INPut]:NULL:CONDition:{U<x>|I<x>}?`
<x> = 1 to 6 (element)

Example `:INPUT:NULL:CONDITION:U1? -> 1`

Description The response is as follows:
0 = NULL feature off
1 = NULL feature in operation

[:INPut]:NULL[:STATe]

Sets or queries the on/off status of the NULL feature.

Syntax `[:INPut]:NULL[:STATe] {<Boolean>}`
[:INPut]:NULL:STATe?

Example `:INPUT:NULL:STATE ON`
`:INPUT:NULL:STATE?`
`-> :INPUT:NULL:STATE 1`

[:INPut]:NULL:TARGet?

Queries all settings for the target of the NULL feature.

Syntax `[:INPut]:NULL:TARGet?`

[:INPut]:NULL:TARGet[:MODE]

Sets or queries the selection mode for the target of the NULL feature.

Syntax `[:INPut]:NULL:TARGet[:MODE] {ALL|SELect}`
[:INPut]:NULL:TARGet:MODE?

Example `:INPUT:NULL:TARGET:MODE ALL`
`:INPUT:NULL:TARGET:MODE?`
`-> :INPUT:NULL:TARGET:MODE ALL`

[:INPut]:NULL:TARGet:{SPEed|TORQue|AUX<x>}

Sets or queries the target of the NULL operation (rotating speed, torque, or AUX).

Syntax `[:INPut]:NULL:TARGet:{SPEed|TORQue|AUX<x>} {ON|HOLD|OFF}`
`[:INPut]:NULL:TARGet:{SPEed|TORQue|AUX<x>}?`
<x> = 1 or 2 (AUX input channel)
ON = NULL feature enabled (When NULL is set to ON, a new NULL value is acquired.)
HOLD = NULL feature enabled (When NULL is set to ON, the previous NULL value is maintained.)
OFF = NULL feature disabled (NULL operation is not performed.)

Example `:INPUT:NULL:TARGET:SPEED ON`
`:INPUT:NULL:TARGET:SPEED?`
`-> :INPUT:NULL:TARGET:SPEED ON`

Description
- SPEed and TORQue are only valid on models with the motor evaluation function (/MTR) option.
- AUX<x> is only valid on models with the auxiliary input (/AUX) option.

[:INPut]:NULL:TARGet:{U<x>|I<x>}

Sets or queries the target of the voltage or current NULL operation of the specified element.

Syntax `[:INPut]:NULL:TARGet:{U<x>|I<x>} {ON|HOLD|OFF}`
`[:INPut]:NULL:TARGet:{U<x>|I<x>}?`
<x> = 1 to 6 (element)
ON = NULL feature enabled (When NULL is set to ON, a new NULL value is acquired.)
HOLD = NULL feature enabled (When NULL is set to ON, the previous NULL value is maintained.)
OFF = NULL feature disabled (NULL operation is not performed.)

Example `:INPUT:NULL:TARGET:U1 ON`
`:INPUT:NULL:TARGET:U1?`
`-> :INPUT:NULL:TARGET:U1 ON`

[:INPut]:NULL:TARGet:{UALL|IALL}

Collectively sets the target of the voltage or current NULL operation of all elements.

Syntax `[:INPut]:NULL:TARGet:{UALL|IALL} {ON|HOLD|OFF}`
ON = NULL feature enabled (When NULL is set to ON, a new NULL value is acquired.)
HOLD = NULL feature enabled (When NULL is set to ON, the previous NULL value is maintained.)
OFF = NULL feature disabled (NULL operation is not performed.)

Example `:INPUT:NULL:TARGET:UALL ON`

[:INPut]:POVer?

Queries the peak over-range information.

Syntax `[:INPut]:POVer?`

Example `:INPUT:POVER? -> 0`

Description
- The peak over-range information of each element is mapped as shown below. For the response, the sum of the values of each bit is returned in decimal format.
- For example, a response of 16 indicates that a peak over-range is occurring at U3.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A2	A1	Tq	Sp	I6	U6	I5	U5	I4	U4	I3	U3	I2	U2	I1	U1

Sp: Rotating speed
Tq: Torque
A1: AUX1
A2: AUX2

[:INPut]:SCALing?

Queries all scaling settings.

Syntax `[:INPut]:SCALing?`

[:INPut]:SCALing:CTPReset?

Queries the CT ratio presets of all elements.

Syntax `[:INPut]:SCALing:CTPReset?`

[:INPut]:SCALing:CTPReset[:ALL]

Collectively sets the CT ratio presets of all elements.

Syntax `[:INPut]:SCALing:CTPReset[:ALL] {OTHERS|CT2000A|CT1000|CT200|CT60}`

Example `:INPUT:SCALING:CTPRESET:ALL CT1000`

Description To use CT1000A, select CT1000.

[:INPut]:SCALing:CTPReset:ELEMent<x>

Sets or queries the CT ratio preset of the specified element.

Syntax `[:INPut]:SCALing:CTPReset:ELEMent<x> {OTHERS|CT2000A|CT1000|CT200|CT60}`
`[:INPut]:SCALing:CTPReset:ELEMent<x>?`
<x> = 1 to 6 (element)

Example `:INPUT:SCALING:CTPRESET:ELEMENT1 CT1000`

Description
- If a preset is queried when the preset value and the CT ratio value are different, "AST" is added to the preset name.
- To use CT1000A, select CT1000.

[:INPut]:SCALing:CTPReset:{SIGMA|SIGMB|SIGMC}

Collectively sets the CT ratio presets of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣB).

Syntax `[:INPut]:SCALing:CTPReset:{SIGMA|SIGMB|SIGMC} {OTHERS|CT2000A|CT1000|CT200|CT60}`

Example `:INPUT:SCALING:CTPRESET:SIGMA CT1000`

Description
- SIGMA, SIGMB, or SIGMC is invalid if the wiring system setting ([:INPut]:WIRing) is made in such a way that the corresponding wiring unit (ΣA, ΣB, or ΣC) does not exist.
- To use CT1000A, select CT1000.

[:INPut]:SCALing:STATe?

Queries the on/off statuses of the scaling of all elements.

Syntax `[:INPut]:SCALing:STATe?`

[:INPut]:SCALing[:STATe][:ALL]

Collectively sets the on/off status of the scaling of all elements.

Syntax `[:INPut]:SCALing[:STATe][:ALL] {<Boolean>}`

Example `:INPUT:SCALING:STATE:ALL OFF`

[:INPut]:SCALing[:STATe]:ELEMent<x>

Sets or queries the on/off status of the scaling of the specified element.

Syntax `[:INPut]:SCALing[:STATe]:ELEMent<x> {<Boolean>}`
[:INPut]:SCALing[:STATe]:ELEMent<x>?
<x> = 1 to 6 (element)

Example
```
:INPUT:SCALING:STATE:ELEMENT1 OFF
:INPUT:SCALING:STATE:ELEMENT1?
-> :INPUT:SCALING:STATE:ELEMENT1 0
```

[:INPut]:SCALing:{VT|CT|SFACtor}?

Queries the VT ratios, CT ratios, or power coefficients of all elements.

Syntax `[:INPut]:SCALing:{VT|CT|SFACtor}?`

[:INPut]:SCALing:{VT|CT|SFACtor}[:ALL]

Collectively sets the VT ratio, CT ratio, or power coefficient of all elements.

Syntax `[:INPut]:SCALing:{VT|CT|SFACtor}[:ALL] {<NRf>}`
<NRf> = 0.0001 to 99999.9999

Example `:INPUT:SCALING:VT:ALL 1`

[:INPut]:SCALing:{VT|CT|SFACtor}:ELEMent<x>

Sets or queries the VT ratio, CT ratio, or power coefficient of the specified element.

Syntax `[:INPut]:SCALing:{VT|CT|SFACtor}:ELEMent<x> {<NRf>}`
`[:INPut]:SCALing:{VT|CT|SFACtor}:ELEMent<x>?`
<x> = 1 to 6 (element)
<NRf> = 0.0001 to 99999.9999

Example
```
:INPUT:SCALING:VT:ELEMENT1 1
:INPUT:SCALING:VT:ELEMENT1?
-> :INPUT:SCALING:VT:ELEMENT1 1.0000
```

[:INPut]:SCALing:{VT|CT|SFACtor}:{SIGMA|SIGMB|SIGMC}

Collectively sets the VT ratio, CT ratio, or power coefficient of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).

Syntax `[:INPut]:SCALing:{VT|CT|SFACtor}:{SIGMA|SIGMB|SIGMC} {<NRf>}`
<NRf> = 0.0001 to 99999.9999

Example `:INPUT:SCALING:VT:SIGMA 1`

Description SIGMA, SIGMB, or SIGMC is invalid if the wiring system setting ([:INPut]:WIRing) is made in such a way that the corresponding wiring unit (ΣA, ΣB, or ΣC) does not exist.

[:INPut]:SYNChronize?

Queries the synchronization sources of all elements.

Syntax `[:INPut]:SYNChronize?`

[:INPut]:SYNChronize[:ALL]

Collectively sets the synchronization source of all elements.

Syntax `[:INPut]:SYNChronize[:ALL]{U<x>|I<x>|EXTernal|NONE}`
<x> = 1 to 6 (element)
EXTernal = External clock input (Ext Clk)
NONE = No synchronization source

Example `:INPUT:SYNCHRONIZE:ALL I1`

[:INPut]:SYNChronize:ELEMent<x>

Sets or queries the synchronization source of the specified element.

Syntax `[:INPut]:SYNChronize:ELEMent<x> {U<x>|I<x>|EXTernal|NONE}`
`[:INPut]:SYNChronize:ELEMent<x>?`
<x> = 1 to 6 (element)
EXTernal = External clock input (Ext Clk)
NONE = No synchronization source

Example
```
:INPUT:SYNCHRONIZE:ELEMENT1 I1
:INPUT:SYNCHRONIZE:ELEMENT1?
-> :INPUT:SYNCHRONIZE:ELEMENT1 I1
```

[:INPut]:SYNChronize:{SIGMA|SIGMB|SIGMC}

Collectively sets the synchronization source of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).

Syntax `[:INPut]:SYNChronize:{SIGMA|SIGMB|SIGMC} {U<x>|I<x>|EXTernal|NONE}`
<x> = 1 to 6 (element)
EXTernal = External clock input (Ext Clk)
NONE = No synchronization source

Example `:INPUT:SYNCHRONIZE:SIGMA I1`

Description SIGMA, SIGMB, or SIGMC is invalid if the wiring system setting ([:INPut]:WIRing) is made in such a way that the corresponding wiring unit (ΣA, ΣB, or ΣC) does not exist.

[:INPut]:SYNChronize:LEVel?

Queries all synchronization source level settings.

Syntax `[:INPut]:SYNChronize:LEVel?`

[:INPut]:SYNChronize:LEVel:{VOLTage|CURRent|EXTSensor}[:ALL]

Sets the synchronization source level of the {voltage|current|external current sensor} of all elements at once.

Syntax `[:INPut]:SYNChronize:LEVel:{VOLTage|CURRent|EXTSensor}[:ALL] {<NRf>}`
<NRf> = -100.0 to 100.0(%)

Example `:INPUT:SYNCHRONIZE:LEVEL:VOLTAGE:ALL 10`

[:INPut]:SYNChronize:LEVel:{VOLTage|CURRent|EXTSensor}:ELEMent<x>

Sets or queries the synchronization source level of the {voltage|current|external current sensor} of the specified element.

Syntax
```
[:INPut]:SYNChronize:LEVel:{VOLTage|
CURRent|EXTSensor}:
ELEMent<x> {<NRf>}
[:INPut]:SYNChronize:LEVel:{VOLTage|
CURRent|EXTSensor}:ELEMent<x>?
```
<x> = 1 to 6 (element)
<NRf> = -100.0 to 100.0(%)

Example
```
:INPUT:SYNCHRONIZE:LEVEL:VOLTAGE:
ELEMNT1 0
:INPUT:SYNCHRONIZE:LEVEL:VOLTAGE:
ELEMNE1?
-> :INPUT:SYNCHRONIZE:LEVEL:VOLTAGE:
ELEMNT1 0.0
```

[:INPut]:SYNChronize:RECTifier?

Queries all synchronization source rectifier on/off settings.

Syntax
```
[:INPut]:SYNChronize:RECTifier?
```

[:INPut]:SYNChronize:RECTifier:{VOLTage|CURRent|EXTSensor}[:ALL]

Sets the synchronization source rectifier on/off setting of the {voltage|current|external current sensor} of all elements at once.

Syntax
```
[:INPut]:SYNChronize: RECTifier:
{VOLTage|CURRent|EXTSensor}[:
ALL] {<Boolean>}
```

Example
```
:INPUT:SYNCHRONIZE:RECTIFIER:
VOLTAGE:ALL OFF
```

[:INPut]:SYNChronize:RECTifier:{VOLTage|CURRent|EXTSensor}:ELEMent<x>

Sets or queries the synchronization source rectifier on/off setting of the {voltage|current|external current sensor} of the specified element.

Syntax
```
[:INPut]:SYNChronize:RECTifier:{VOLTage|
CURRent|EXTSensor}:
ELEMent<x> {<Boolean>}
[:INPut]:SYNChronize:RECTifier:
{VOLTage|CURRent|EXTSensor}:
ELEMent<x>?
```
<x> = 1 to 6 (element)

Example
```
:INPUT:SYNCHRONIZE:RECTIFIER:
VOLTAGE:ELEMNT1 OFF
:INPUT:SYNCHRONIZE:RECTIFIER:
VOLTAGE:ELEMNE1?
-> :INPUT:SYNCHRONIZE:RECTIFIER:
VOLTAGE:ELEMNT1 0
```

[:INPut]:VOLTage?

Queries all voltage measurement settings.

Syntax
```
[:INPut]:VOLTage?
```

[:INPut]:VOLTage:AUTO?

Queries the voltage auto range on/off statuses of all elements.

Syntax
```
[:INPut]:VOLTage:AUTO?
```

[:INPut]:VOLTage:AUTO[:ALL]

Collectively sets the voltage auto range on/off status of all elements.

Syntax
```
[:INPut]:VOLTage:AUTO[:ALL] {<Boolean>}
```

Example
```
:INPUT:VOLTAGE:AUTO:ALL ON
```

[:INPut]:VOLTage:AUTO:ELEMent<x>

Sets or queries the voltage auto range on/off status of the specified element.

Syntax
```
[:INPut]:VOLTage:AUTO:
ELEMent<x> {<Boolean>}
[:INPut]:VOLTage:AUTO:ELEMent<x>?
```
<x> = 1 to 6 (element)

Example
```
:INPUT:VOLTAGE:AUTO:ELEMENT1 ON
:INPUT:VOLTAGE:AUTO:ELEMENT1?
-> :INPUT:VOLTAGE:AUTO:ELEMENT1 1
```

[:INPut]:VOLTage:AUTO:{SIGMA|SIGMB|SIGMC}

Collectively sets the voltage auto range on/off status of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).

Syntax
```
[:INPut]:VOLTage:AUTO:{SIGMA|SIGMB|
SIGMC} {<Boolean>}
```

Example
```
:INPUT:VOLTAGE:AUTO:SIGMA ON
```

Description SIGMA, SIGMB, or SIGMC is invalid if the wiring system setting ([:INPut]:WIRing) is made in such a way that the corresponding wiring unit (ΣA, ΣB, or ΣC) does not exist.

[:INPut]:VOLTage:CONFig?

Queries the valid voltage ranges of all elements.

Syntax
```
[:INPut]:VOLTage:CONFig?
```

[:INPut]:VOLTage:CONFig[:ALL]

Collectively sets the valid voltage range of all elements.

Syntax `[:INPut]:VOLTage:CONFig[:ALL] {ALL|<Voltage>[,<Voltage>][,<Voltage>]...}`

ALL = All ranges are valid.

- When the crest factor is set to 3:
  <Voltage> = 1.5 V, 3 V, 6 V, 10 V, 15 V, 30 V, 60 V, 100 V, 150 V, 300 V, 600 V, 1000 V
- When the crest factor is set to 6 or 6A:
  <Voltage> = 0.75 V, 1.5 V, 3 V, 5 V, 7.5 V, 15 V, 30 V, 50 V, 75 V, 150 V, 300 V, 500 V

Example
```
:INPUT:VOLTAGE:CONFIG:ALL ALL
:INPUT:VOLTAGE:CONFIG:ALL 1000,300,100,30,10
```

Description In the parameters, list all the voltage ranges that you want to enable. To enable all the ranges, specify the parameter “ALL.”

[:INPut]:VOLTage:CONFig:ELEMent<x>

Sets or queries the valid voltage ranges of the specified element.

Syntax
```
[:INPut]:VOLTage:CONFig:ELEMent<x>
{ALL|<Voltage>[,<Voltage>][,<Voltage>]...}
[:INPut]:VOLTage:CONFig:ELEMent<x>?
```
<x> = 1 to 6 (element)
ALL = All ranges are valid.
<Voltage> = See [:INPut]:VOLTage:CONFig[:ALL]

Example
```
:INPUT:VOLTAGE:CONFIG:ELEMENT1 ALL
:INPUT:VOLTAGE:CONFIG:ELEMENT1?
-> :INPUT:VOLTAGE:CONFIG:ELEMENT1 ALL
:INPUT:VOLTAGE:CONFIG:ELEMENT1 1000,300,100,30,10
:INPUT:VOLTAGE:CONFIG:ELEMENT1?
-> :INPUT:VOLTAGE:CONFIG:ELEMENT1 1.0000E+03,300.0E+00,100.0E+00,30.0E+00,10.0E+00
```

Description In the parameters, list all the voltage ranges that you want to enable. To enable all the ranges, specify the parameter “ALL.”

[:INPut]:VOLTage:POJump?

Queries the jump destination ranges of all elements that are used when a voltage peak over-range occurs.

Syntax `[:INPut]:VOLTage:POJump?`

[:INPut]:VOLTage:POJump[:ALL]

Collectively sets the jump destination range of all elements that is used when a voltage peak over-range occurs.

Syntax `[:INPut]:VOLTage:POJump[:ALL]{OFF|<Voltage>}`

OFF = No jump destination voltage range

- When the crest factor is set to 3:
  <Voltage> = 1.5 V, 3 V, 6 V, 10 V, 15 V, 30 V, 60 V, 100 V, 150 V, 300 V, 600 V, 1000 V
- When the crest factor is set to 6 or 6A:
  <Voltage> = 0.75 V, 1.5 V, 3 V, 5 V, 7.5 V, 15 V, 30 V, 50 V, 75 V, 150 V, 300 V, 500 V

Example
```
:INPUT:VOLTAGE:POJUMP:ALL OFF
```

[:INPut]:VOLTage:POJump:ELEMent<x>

Sets or queries the jump destination range of the specified element that is used when a voltage peak over-range occurs.

Syntax
```
[:INPut]:VOLTage:POJump:
ELEMent<x> {OFF|<Voltage>}
[:INPut]:VOLTage:POJump:ELEMent<x>?
```
<x> = 1 to 6 (element)
OFF = No jump destination voltage range
<Voltage> = See [:INPut]:VOLTage:POJump[:ALL]

Example
```
:INPUT:VOLTAGE:POJUMP:ELEMENT1 1000V
:INPUT:VOLTAGE:POJUMP:ELEMENT1?
-> :INPUT:VOLTAGE:POJUMP:
ELEMENT1 1.0000E+03
```

[:INPut]:VOLTage:RANGe?

Queries the voltage ranges of all elements.

Syntax `[:INPut]:VOLTage:RANGe?`

[:INPut]:VOLTage:RANGe[:ALL]

Collectively sets the voltage range of all elements.

Syntax
```
[:INPut]:VOLTage:RANGe[:
ALL] {<Voltage>}
```

- When the crest factor is set to 3:
  <Voltage> = 1.5 V, 3 V, 6 V, 10 V, 15 V, 30 V, 60 V, 100 V, 150 V, 300 V, 600 V, 1000 V
- When the crest factor is set to 6 or 6A:
  <Voltage> = 0.75 V, 1.5 V, 3 V, 5 V, 7.5 V, 15 V, 30 V, 50 V, 75 V, 150 V, 300 V, 500 V

Example
```
:INPUT:VOLTAGE:RANGE:ALL 1000V
```

[:INPut]:VOLTage:RANGe:ELEMent<x>

Sets or queries the voltage range of the specified element.

Syntax
```
[:INPut]:VOLTage:RANGe:
ELEMent<x> {<Voltage>}
[:INPut]:VOLTage:RANGe:ELEMent<x>?
```
<x> = 1 to 6 (element)
<Voltage> = See [:INPut]:VOLTage:RANGe[:ALL]

Example
```
:INPUT:VOLTAGE:RANGE:ELEMENT1 1000V
:INPUT:VOLTAGE:RANGE:ELEMENT1?
-> :INPUT:VOLTAGE:RANGE:
ELEMENT1 1.000E+03
```

[:INPut]:VOLTage:RANGe:{SIGMA|SIGMB|SIGMC}

Collectively sets the voltage range of all the elements that belong to the specified wiring unit (ΣA, ΣB, or ΣC).

Syntax `[:INPut]:VOLTage:RANGe:{SIGMA|SIGMB|SIGMC} {<Voltage>}`
<Voltage> = See [:INPut]:VOLTage:RANGe[:ALL]]

Example `:INPUT:VOLTAGE:RANGE:SIGMA 1000V`

Description SIGMA, SIGMB, or SIGMC is invalid if the wiring system setting ([:INPut]:WIRing) is made in such a way that the corresponding wiring unit (ΣA, ΣB, or ΣC) does not exist.

[:INPut]:WIRing

Sets or queries the wiring system.

Syntax
```
[:INPut]:WIRing {(P1W2|P1W3|P3W3|P3W4|V3A3)[,(P1W2|P1W3|P3W3|P3W4|V3A3)][,(P1W2|P1W3|P3W3|P3W4|V3A3)][,(P1W2|P1W3|P3W3|P3W4|V3A3)][,(P1W2|P1W3|P3W3)][,P1W2]}
[:INPut]:WIRing?
```
P1W2 = Single-phase, two-wire system [1P2W]
P1W3 = Single-phase, three-wire system [1P3W]
P3W3 = Three-phase, three-wire system [3P3W]
P3W4 = Three-phase, four-wire system [3P3W]
V3A3 = Three-phase, three-wire system with a three-voltage, three-current method [3P3W(3V3A)]

Example
- Example for a 6-element model
```
:INPUT:WIRING P1W2,P1W2,P1W2,P1W2,P1W2,P1W2
:INPUT:WIRING?
-> :INPUT:WIRING P1W2,P1W2,P1W2,P1W2,P1W2,P1W2
:INPUT:WIRING P1W3,P1W3,P1W3
:INPUT:WIRING?
-> :INPUT:WIRING P1W3,P1W3,P1W3
:INPUT:WIRING P3W4,V3A3
:INPUT:WIRING?
-> :INPUT:WIRING P3W4,V3A3
```
- Example for a 3-element model
```
:INPUT:WIRING P1W2,P3W3
:INPUT:WIRING?
-> :INPUT:WIRING P1W2,P3W3
:INPUT:WIRING P3W4
:INPUT:WIRING?
-> :INPUT:WIRING P3W4
```

Description
- Set the wiring system pattern in order starting from the element with the smallest number.
- Some wiring system patterns cannot be selected on certain model types. For details on the available wiring system patterns, see the features guide, IM WT1801R-01EN.
- The pattern is fixed to P1W2 on 1-element models. No other setting is allowed.

5.14 INTEGrate Group

The commands in this group deal with integration.
You can make the same settings and queries that you can make by pressing INTEG on the front panel.

:INTEGrate?

Queries all integration settings.

Syntax :INTEGrate?

:INTEGrate:ACAL

Sets or queries the on/off status of integration auto calibration.

Syntax :INTEGrate:ACAL {<Boolean>}
:INTEGrate:ACAL?

Example :INTEGRATE:ACAL OFF
:INTEGRATE:ACAL?
-> :INTEGRATE:ACAL 0

:INTEGrate:INDependent

Sets or queries the on/off status of independent element integration.

Syntax :INTEGrate:INDependent {<Boolean>}
:INTEGrate:INDependent?

Example :INTEGRATE:INDEPENDENT OFF
:INTEGRATE:INDEPENDENT?
-> :INTEGRATE:INDEPENDENT 0

:INTEGrate:MODE

Sets or queries the integration mode.

Syntax :INTEGrate:MODE {NORMal|CONTinuous|RNORmal|RCONtinuous}
:INTEGrate:MODE?
NORMal = Normal integration mode
CONTinuous = Continuous integration mode
RNORmal = Real-time normal integration mode
RCONtinuous = Real-time continuous integration mode

Example :INTEGRATE:MODE NORMAL
:INTEGRATE:MODE?
-> :INTEGRATE:MODE NORMAL

:INTEGrate:QMODe?

Queries the electric current modes for electric current integration of all elements.

Syntax :INTEGrate:QMODe?

:INTEGrate:QMODe[:ALL]

Collectively sets the electric current mode for electric current integration of all elements.

Syntax :INTEGrate:QMODe[:ALL] {RMS|MEAN|DC|RMEAN|AC}

Example :INTEGRATE:QMODE:ALL DC

:INTEGrate:QMODe:ELEMent<x>

Sets or queries the electric current mode for electric current integration of the specified element.

Syntax :INTEGrate:QMODe:ELEMent<x> {RMS|MEAN|DC|RMEAN|AC}
:INTEGrate:QMODe:ELEMent<x>?
<x> = 1 to 6 (element)

Example :INTEGRATE:QMODE:ELEMENT1 DC
:INTEGRATE:QMODE:ELEMENT1?
-> :INTEGRATE:QMODE:ELEMENT1 DC

Description Regardless of the independent element integration setting (:INTEGrate:INDependent), this instrument operates according to the electric current mode of the specified element.

:INTEGrate:RACTion

Sets or queries the integration resume action that is taken when the power recovers from a power failure while integration is in progress.

Syntax :INTEGrate:RACTion {STARt|STOP|ERRor}
:INTEGrate:RACTion?
STARt = Integration is resumed automatically.
STOP = Integration is placed in a stopped state. Integration can be resumed manually (START key or command).
ERRor = Integration error state occurs. Resuming of integration is prohibited.

Example :INTEGRATE:RATION STOP
:INTEGRATE:RACTION?
-> :INTEGRATE:RACTION STOP

:INTEGrate:RESet

Resets the integrated value.

Syntax :INTEGrate:RESet {[<NRf>][,<NRf>][,<NRf>][,<NRf>][,<NRf>][,<NRf>]}
<NRf> = 1 to 6 (element)

Example :INTEGRATE:RESET (resets all elements)
:INTEGRATE:RESET 1,2,3 (resets the specified elements)

Description
- When independent element integration (:INTEGrate:INDependent) is set to ON (1), you can use the parameters to specify which elements to perform the operation on. If you omit the parameters, the operation will be performed on all the elements.
- When independent element integration (:INTEGrate:INDependent) is set to OFF (0), you cannot use the parameters to specify elements.

:INTEGrate:RTALl:{STARt|END}

Collectively sets the integration start or end time of all elements for real-time integration mode.

Syntax :INTEGrate:RTALl:{STARt|END} {<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>}
{<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>} = 2001,1,1,0,0,0 to 2099,12,31,23,59,59
First <NRf> = 2001 to 2099 (year)
Second <NRf> = 1 to 12 (month)
Third <NRf> = 1 to 31 (day)
Fourth <NRf> = 0 to 23 (hour)
Fifth <NRf> = 0 to 59 (minute)
Sixth <NRf> = 0 to 59 (second)

Example :INTEGRATE:RTALL:START 2024,1,1,0,0,0

:INTEGrate:RTIMe<x>?

Queries the integration start and end times for real-time integration mode.

Syntax :INTEGrate:RTIMe<x>?
<x> = 1 to 6 (element)

:INTEGrate:RTIMe<x>:{STARt|END}

Sets or queries the integration start or end time for real-time integration mode.

Syntax :INTEGrate:RTIMe<x>:{STARt|END} {<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>}
:INTEGrate:RTIMe<x>:{STARt|END}?
<x> = 1 to 6 (element)
{<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>} = 2001,1,1,0,0,0 to 2099,12,31,23,59,59
First <NRf> = 2001 to 2099 (year)
Second <NRf> = 1 to 12 (month)
Third <NRf> = 1 to 31 (day)
Fourth <NRf> = 0 to 23 (hour)
Fifth <NRf> = 0 to 59 (minute)
Sixth <NRf> = 0 to 59 (second)

Example :INTEGRATE:RTIME1:START 2024,1,1,0,0,0
:INTEGRATE:RTIME1:START?
-> :INTEGRATE:RTIME1:START 2024,1,1,0,0,0

Description
- This command is valid when the integration mode (:INTEGrate:MODE) is set to real-time integration mode (RNORmal or RCONtinuous).
- When independent element integration (:INTEGrate:INDependent) is set to OFF (0), you can omit <x> (<x> = 1). The operation is performed on all elements using element 1's integration start or end time.

:INTEGrate:STARt

Starts integration.

Syntax :INTEGrate:STARt {[<NRf>][,<NRf>][,<NRf>][,<NRf>][,<NRf>][,<NRf>]}
<NRf> = 1 to 6 (element)

Example :INTEGRATE:START (starts integration on all elements)
:INTEGRATE:START 1,2,3 (starts integration on the specified elements)

Description
- When independent element integration (:INTEGrate:INDependent) is set to ON (1), you can use the parameters to specify which elements to perform the operation on. If you omit the parameters, the operation will be performed on all the elements.
- When independent element integration (:INTEGrate:INDependent) is set to OFF (0), you cannot use the parameters to specify elements.

:INTEGrate:STATe?

Queries the integration status.

Syntax :INTEGrate:STATe? {<NRf>}
:INTEGrate:STATe?
<NRf> = 1 to 6 (element)

Example
- When independent element integration (:INTEGrate:INDependent) is set to OFF (0):
  :INTEGRATE:STATE? -> RESET
- When independent element integration (:INTEGrate:INDependent) is set to ON (1):
  :INTEGRATE:STATE? 1 -> RESET
  :INTEGRATE:STATE?
  -> RESET,RESET,RESET,RESET,RESET,RESET

Description
- The response is as follows:
  RESet = Integration reset
  READy = Waiting (real-time integration mode)
  STARt = Integration in progress
  STOP = Integration stop
  ERRor = Abnormal integration termination (integration overflow, power failure)
  TIMeup = Integration stop due to integration timeout
- When independent element integration (:INTEGrate:INDependent) is set to OFF (0), you cannot use the parameter to specify the element that you want to query.
- When independent element integration (:INTEGrate:INDependent) is set to ON (1), you can use the parameter to specify the element. If the parameter is omitted, the integration statuses of all elements are output in order, starting with element 1.

:INTEGrate:STOP

Stops integration.

Syntax :INTEGrate:STOP {[<NRf>][,<NRf>][,<NRf>][,<NRf>][,<NRf>][,<NRf>]}
<NRf> = 1 to 6 (element)

Example :INTEGRATE:STOP (stops integration on all elements)
:INTEGRATE:STOP 1,2,3 (stops integration on the specified elements)

Description
- When independent element integration (:INTEGrate:INDependent) is set to ON (1), you can use the parameters to specify which elements to perform the operation on. If you omit the parameters, the operation will be performed on all the elements.
- When independent element integration (:INTEGrate:INDependent) is set to OFF (0), you cannot use the parameters to specify elements.

:INTEGrate:TIMer<x>

Sets or queries the integration timer value.

Syntax :INTEGrate:TIMer<x> {<NRf>,<NRf>,<NRf>}
:INTEGrate:TIMer<x>?
<x> = 1 to 6 (element)
{<NRf>,<NRf>,<NRf>} = 0, 0, 0 to 10000, 0, 0
First <NRf> = 0 to 10000 (hours)
Second <NRf> = 0 to 59 (minutes)
Third <NRf> = 0 to 59 (seconds)

Example :INTEGRATE:TIMER1 1,0,0
:INTEGRATE:TIMER1?
-> :INTEGRATE:TIMER1 1,0,0

Description When independent element integration (:INTEGrate:INDependent) is set to OFF (0), you can omit <x> (<x> = 1). The operation is performed on all elements using element 1's integration timer.

:INTEGrate:TMALl

Collectively sets the integration timer of all elements.

Syntax :INTEGrate:TMALl {<NRf>,<NRf>,<NRf>}
{<NRf>,<NRf>,<NRf>} = 0, 0, 0 to 10000, 0, 0
First <NRf> = 0 to 10000 (hours)
Second <NRf> = 0 to 59 (minutes)
Third <NRf> = 0 to 59 (seconds)

Example :INTEGRATE:TMALL 1,0,0

:INTEGrate:WPTYpe?

Queries the watt-hour integration methods for each polarity (WP+/WP–) of all elements.

Syntax :INTEGrate:WPTYpe?

:INTEGrate:WPTYpe[:ALL]

Collectively sets the watt-hour integration method for each polarity (WP+/WP–) of all elements.

Syntax :INTEGrate:WPTYpe[:ALL] {CHARge|SOLD}

Example :INTEGRATE:WPTYPE:ALL CHARGE

:INTEGrate:WPTYpe:ELEMent<x>

Sets or queries the watt-hour integration method for each polarity (WP+/WP–) of the specified element.

Syntax :INTEGrate:WPTYpe:ELEMent<x> {CHARge|SOLD}
:INTEGrate:WPTYpe:ELEMent<x>?
<x> = 1 to 6 (element)
CHARge = Charge-discharge
SOLD = Commercial power

Example :INTEGRATE:WPTYPE:ELEMENT1 CHARGE
:INTEGRATE:WPTYPE:ELEMENT1?
-> :INTEGRATE:WPTYPE:ELEMENT1 CHARGE

Description Regardless of the independent element integration setting (:INTEGrate:INDependent), this instrument operates according to the integration method of the specified element.

5.15 MEASure Group

The commands in this group deal with computation.
You can make the same settings and queries that you can make by pressing MEASURE, and AVG on the front panel or by pressing WIRING on the front panel and then using the η Formula or Δ Measure menu.

:MEASure?

Queries all computation settings.

Syntax :MEASure?

:MEASure:AVERaging?

Queries all averaging settings.

Syntax :MEASure:AVERaging?

:MEASure:AVERaging:COUNt

Sets or queries the averaging coefficient.

Syntax :MEASure:AVERaging:COUNt {<NRf>}
:MEASure:AVERaging:COUNt?
<NRf> = 2 to 64 (attenuation constant when TYPE = EXPonent)
<NRf> = 8 to 64 (moving average count when TYPE = LINear)

Example :MEASURE:AVERAGING:COUNT 2
:MEASURE:AVERAGING:COUNT?
-> :MEASURE:AVERAGING:COUNT 2

Description The averaging of harmonic measurement functions (option) is only valid when TYPE is set to EXPonent (attenuation constant). For details, see the features guide, IM WT1801R-01EN.

:MEASure:AVERaging[:STATe]

Sets or queries the on/off status of averaging.

Syntax :MEASure:AVERaging[:STATe] {<Boolean>}
:MEASure:AVERaging:STATe?

Example :MEASURE:AVERAGING:STATE ON
:MEASURE:AVERAGING:STATE?
-> :MEASURE:AVERAGING:STATE 1

:MEASure:AVERaging:TYPE

Sets or queries the averaging type.

Syntax :MEASure:AVERaging:TYPE {EXPonent|LINear}
:MEASure:AVERaging:TYPE?

Example :MEASURE:AVERAGING:TYPE EXPONENT
:MEASURE:AVERAGING:TYPE?
-> :MEASURE:AVERAGING:TYPE EXPONENT

Description The averaging of harmonic measurement functions (option) is only valid when the type is set to EXPonent. For details, see the features guide, IM WT1801R-01EN.

:MEASure:DMeasure?

Queries all delta computation settings.

Syntax :MEASure:DMeasure?

:MEASure:DMeasure:MODE

Sets or queries the voltage or current mode that is used in delta computation.

Syntax :MEASure:DMeasure:MODE {RMS|MEAN|DC|RMEAN|AC}
:MEASure:DMeasure:MODE?

Example :MEASURE:DMEASURE:MODE RMS
:MEASURE:DMEASURE:MODE?
-> :MEASURE:DMEASURE:MODE RMS

:MEASure:DMeasure:{SIGMA|SIGMB|SIGMC}

Sets or queries the delta computation mode for wiring unit ΣA, ΣB, or ΣC.

Syntax :MEASure:DMeasure:{SIGMA|SIGMB|SIGMC} {OFF|DIFFerence|P3W3_V3A3|ST_DT|DT_ST}
:MEASure:DMeasure:{SIGMA|SIGMB|SIGMC}?

Example :MEASURE:DMEASURE:SIGMA OFF
:MEASURE:DMEASURE:SIGMA?
-> :MEASURE:DMEASURE:SIGMA OFF

Description The available options are explained below. The modes that can be selected vary depending on the wiring system of the specified wiring unit (ΣA, ΣB, or ΣC).
OFF = No delta computation (only selectable with a single-phase, two-wire system—1P2W)
DIFFerence = Differential voltage, differential current (only selectable with a single-phase, three-wire system—1P3W—or a three-phase, three-wire system—3P3W)
P3W3_V3A3 = 3P3W-to-3V3A conversion (only selectable with a single-phase, three-wire system—1P3W—or a three-phase, three-wire system—3P3W)
ST_DT = Star-to-delta conversion (only selectable with a three-phase, four-wire system—3P4W)
DT_ST = Delta-to-star conversion (only selectable with a three-phase, three-wire system that uses the three-voltage, three current method—3P3W(3V3A))

:MEASure:EFFiciency?

Queries all efficiency computation settings.

Syntax :MEASure:EFFiciency?

:MEASure:EFFiciency:ETA<x>

Sets or queries the efficiency equation.

Syntax :MEASure:EFFiciency:ETA<x> {(OFF|P<x>|PA|PB|PC|PM|UDEF<x>)[,(OFF|P<x>|PA|PB|PC|PM|UDEF<x>)]}
:MEASure:EFFiciency:ETA<x>?
ETA<x>'s <x> = 1 to 4 (η1 to η4)
OFF = No computation
P<x>'s <x> = 1 to 6 (element)
PA, PB, PC = PΣA, PΣB, PΣC (the available options vary depending on the number of elements)
PM = Pm (motor output, only on models with the motor evaluation function [/MTR] option)
UDEF<x>'s <x> = 1 or 2 (Udef1 or Udef2)

Example :MEASURE:EFFICIENCY:ETA1 P3,PA
:MEASURE:EFFICIENCY:ETA1?
-> :MEASURE:EFFICIENCY:ETA1 P3,PA

Description
- Set the numerator and then the denominator.
- The denominator can be omitted. The denominator is set to OFF when it is omitted.
- The denominator is omitted from the response to a query when it is OFF.

:MEASure:EFFiciency:UDEF<x>

Sets or queries the user-defined parameters used in the efficiency equation.

Syntax :MEASure:EFFiciency:UDEF<x> {(NONE|P<x>|PA|PB|PC|PM)[,(NONE|P<x>|PA|PB|PC|PM)][,(NONE|P<x>|PA|PB|PC|PM)]}
:MEASure:EFFiciency:UDEF<x>?
UDEF<x>'s <x> = 1 or 2 (Udef1 or Udef2)
NONE = No operand
P<x>'s <x> = 1 to 6 (element)
PA, PB, PC = PΣA, PΣB, PΣC (the available options vary depending on the number of elements)
PM = Pm (motor output, only on models with the motor evaluation function [/MTR] option)

Example :MEASURE:EFFICIENCY:UDEF1 P1,P2,P3
:MEASURE:EFFICIENCY:UDEF1?
-> :MEASURE:EFFICIENCY:UDEF1 P1,P2,P3

Description
- Set the parameters in ascending order.
- Parameters 2 to 4 can be omitted. Omitted parameters are set to NONE.
- Parameters 2 to 4 are omitted from the response to a query if all the subsequent parameters are NONE.

:MEASure:EVENt<x>?

Queries all the settings of the specified user-defined event.

Syntax :MEASure:EVENt<x>?
<x> = 1 to 8 (Event1 to Event8)

:MEASure:EVENt<x>:EXPRession?

Queries all the settings of the specified user-defined event's expression.

Syntax :MEASure:EVENt<x>:EXPRession?
<x> = 1 to 8 (Event1 to Event8)

:MEASure:EVENt<x>:EXPRession:CONDition

Sets or queries the specified user-defined event's expression (compound condition type).

Syntax :MEASure:EVENt<x>:EXPRession:CONDition {<Event>[,<Logic>,<Event>][,<Logic>,<Event>]...}
:MEASure:EVENt<x>:EXPRession:CONDition?
<x> = 1 to 8 (Event1 to Event8)
<Event> = {<NRf>} (<NRf) = 1 to 8 (Event1 to Event8))
<Logic> = {AND|OR}

Example :MEASURE:EVENT1:EXPRESSION:CONDITION 1,AND,2
:MEASURE:EVENT1:EXPRESSION:CONDITION?
-> :MEASURE:EVENT1:EXPRESSION:CONDITION 1,AND,2

Description This command is valid when the expression type (:MEASure:EVENt<x>:EXPRession:TYPE) is set to CONDition.

:MEASure:EVENt<x>:EXPRession:INVerse

Sets or queries the on/off status of the logic inversion of the specified user-defined event's expression (compound condition type).

Syntax :MEASure:EVENt<x>:EXPRession:INVerse {<Boolean>}
:MEASure:EVENt<x>:EXPRession:INVerse?
<x> = 1 to 8 (Event1 to Event8)

Example :MEASURE:EVENT1:EXPRESSION:INVERSE OFF
:MEASURE:EVENT1:EXPRESSION:INVERSE?
-> :MEASURE:EVENT1:EXPRESSION:INVERSE 0

Description This command is valid when the expression type (:MEASure:EVENt<x>:EXPRession:TYPE) is set to CONDition.

:MEASure:EVENt<x>:EXPRession:ITEM

Sets or queries the target item of the specified user-defined event's expression (range-defined type).

Syntax
```
:MEASure:EVENt<x>:EXPRession:
ITEM {<Function>[,<Element>]
[,<Order>]}
:MEASure:EVENt<x>:EXPRession:ITEM?
```
<x> = 1 to 8 (Event1 to Event8)
<Function> = {URMS|IRMS|P|S|Q|...}
<Element> = {<NRf>|SIGMa|SIGMB|SIGMC}
(<NRf> = 1 to 6)
<Order> = {TOTal|DC|<NRf>}
(<NRf> = 1 to 500)

Example
```
:MEASURE:EVENT1:EXPRESSION:
ITEM URMS,1
:MEASURE:EVENT1:EXPRESSION:ITEM?
-> :MEASURE:EVENT1:EXPRESSION:
ITEM URMS,1
```

Description
- This command is valid when the expression type (:MEASure:EVENt<x>:EXPRession:TYPE) is set to RANGe.
- For information about the options available for <Function>, see "Numeric data functions" in "Function option list" at the end of the DISPlay group on page 5-44.
- If <Element> is omitted, the element is set to 1.
- If <Order> is omitted, the order is set to TOTal.
- <Element> and <Order> are omitted from responses to functions that do not need them.

:MEASure:EVENt<x>:EXPRession:LIMit<x>

Sets or queries the range of the specified user-defined event's expression (range-defined type).

Syntax
```
:MEASure:EVENt<x>:EXPRession:LIMit<x>
{<Operand>,<NRf>}
:MEASure:EVENt<x>:EXPRession:
LIMit<x>?
```
EVENt<x>'s <x> = 1 to 8 (Event1 to Event8)
LIMit<x>'s <s> = 1 or 2
<Operand> = {OFF|LESS|LEQual|EQual|GReat|GEQual|NEQual}
<NRf> = -9.9999E+12 to 9.9999E+12

Example
```
:MEASURE:EVENT1:EXPRESSION:
LIMIT1 LESS,100
:MEASURE:EVENT1:EXPRESSION:LIMIT1?
-> :MEASURE:EVENT1:EXPRESSION:
LIMIT1 LESS,100.00E+00
:MEASURE:EVENT1:EXPRESSION:LIMIT2 OFF
:MEASURE:EVENT1:EXPRESSION:LIMIT2?
-> :MEASURE:EVENT1:EXPRESSION:LIM
IT2 OFF
```

Description
- This command is valid when the expression type (:MEASure:EVENt<x>:EXPRession:TYPE) is set to RANGe.
- When <Operand> is set to OFF, <NRf> can be omitted.

:MEASure:EVENt<x>:EXPRession:STRing?

Queries the specified user-defined event's expression as a string.

Syntax
```
:MEASure:EVENt<x>:EXPRession:STRing?
```
<x> = 1 to 8 (Event1 to Event8)

Example
```
:MEASURE:EVENT1:EXPRESSION:STRING?
-> "TEMP < 100.00000"
```

Description If there is no expression, "No Expression" is returned.

:MEASure:EVENt<x>:EXPRession:TYPE

Sets or queries the specified user-defined event's expression type.

Syntax
```
:MEASure:EVENt<x>:EXPRession:
TYPE {RANGe|CONDition}
:MEASure:EVENt<x>:EXPRession:TYPE?
```
<x> = 1 to 8 (Event1 to Event8)
RANGe = Range-defined type
CONDition = Compound event type

Example
```
:MEASURE:EVENT1:EXPRESSION:
TYPE RANGE
:MEASURE:EVENT1:EXPRESSION:TYPE?
-> :MEASURE:EVENT1:EXPRESSION:
TYPE RANGE
```

:MEASure:EVENt<x>:FLABel

Sets or queries the string that is displayed when the specified user-defined event's condition is not met.

Syntax
```
:MEASure:EVENt<x>:FLABel {<String>}
:MEASure:EVENt<x>:FLABel?
```
<x> = 1 to 8 (Event1 to Event8)
<String> = Up to 6 characters

Example
```
:MEASURE:EVENT1:FLABEL "False"
:MEASURE:EVENT1:FLABEL?
-> :MEASURE:EVENT1:FLABEL "False"
```

:MEASure:EVENt<x>:NAME

Sets or queries the specified user-defined event's name.

Syntax
```
:MEASure:EVENt<x>:NAME {<String>}
:MEASure:EVENt<x>:NAME?
```
<x> = 1 to 8 (Event1 to Event8)
<String> = Up to 8 characters

Example
```
:MEASURE:EVENT1:NAME "Ev1"
:MEASURE:EVENT1:NAME?
-> :MEASURE:EVENT1:NAME "Ev1"
```

:MEASure:EVENt<x>[:STATe]

Sets or queries the on/off status of the specified user-defined event.

Syntax
```
:MEASure:EVENt<x>[:STATe] {<Boolean>}
:MEASure:EVENt<x>:STATe?
```
<x> = 1 to 8 (Event1 to Event8)

Example
```
:MEASURE:EVENT1:STATE ON
:MEASURE:EVENT1:STATE?
-> :MEASURE:EVENT1:STATE 1
```

:MEASure:EVENt<x>:TLABel

Sets or queries the string that is displayed when the specified user-defined event's condition is met.

Syntax
```
:MEASure:EVENt<x>:TLABel {<String>}
:MEASure:EVENt<x>:TLABel?
```
<x> = 1 to 8 (Event1 to Event8)
<String> = Up to 6 characters

Example
```
:MEASURE:EVENT1:TLABEL "True"
:MEASURE:EVENT1:TLABEL?
-> :MEASURE:EVENT1:TLABEL "True"
```

:MEASure:FUNCtion<x>?

Queries all the settings of the specified user-defined function.

Syntax
```
:MEASure:FUNCtion<x>?
```
<x> = 1 to 20 (F1 to F20)

:MEASure:FUNCtion<x>:EXPRession

Sets or queries the equation of the specified user-defined function.

Syntax
```
:MEASure:FUNCtion<x>:
EXPRession {<String>}
:MEASure:FUNCtion<x>:EXPRession?
```
<x> = 1 to 20 (F1 to F20)
<String> = Up to 60 characters

Example
```
:MEASURE:FUNCTION1:
EXPRESSION "WH(E1)/TI(E1)*3600"
:MEASURE:FUNCTION1:EXPRESSION?
-> :MEASURE:FUNCTION1:
EXPRESSION "WH(E1)/TI(E1)*3600"
```

:MEASure:FUNCtion<x>:NAME

Sets or queries the name of the specified user-defined function.

Syntax
```
:MEASure:FUNCtion<x>:NAME {<String>}
:MEASure:FUNCtion<x>:NAME?
```
<x> = 1 to 20 (F1 to F20)
<String> = Up to 8 characters

Example
```
:MEASURE:FUNCTION1:NAME "F1"
:MEASURE:FUNCTION1:NAME?
-> :MEASURE:FUNCTION1:NAME "F1"
```

:MEASure:FUNCtion<x>:PRESet[:EXECute]

Executes a user-defined function preset.

Syntax
```
:MEASure:FUNCtion<x>:
PRESet[:EXECute] {DQP3W4|DQV3A3|
REL_DQP3W4|REL_DQV3A3|P6W7}
```
<x> = 1 to 20 (F1 to F20)
* The input value is ignored. The operation is applied to all functions.

Example
```
:MEASURE:FUNCTION1:PRESET:
EXECUTE DQV3A3
```

Description
- A user-defined function preset is executed.
- The <x> value in FUNCtion<x> has no meaning in the setting.

:MEASure:FUNCtion<x>:PRESet:FILE:SAVE

Saves the contents of user-defined functions to a file.

Syntax
```
:MEASure:FUNCtion<x>:PRESet:FILE:
SAVE {<String>}
```
<x> = 1 to 20 (F1 to F20)
* The input value is ignored. The operation is applied to all functions.
<String> = File name

Example
```
:MEASURE:FUNCTION1:PRESET:FILE:
SAVE "UserFunc1"
```

Description
- A user-defined function file is saved.
- The <x> value in FUNCtion<x> has no meaning in the setting.

:MEASure:FUNCtion<x>:PRESet:FILE:LOAD

Loads a user-defined function file.

Syntax
```
:MEASure:FUNCtion<x>:PRESet:FILE:
LOAD {<String>}
```
<x> = 1 to 20 (F1 to F20)
* The input value is ignored. The operation is applied to all functions.
<String> = File name

Example
```
:MEASURE:FUNCTION1:PRESET:FILE:
LOAD "UserFunc1"
```

Description
- A user-defined function file is loaded.
- The <x> value in FUNCtion<x> has no meaning in the setting.

:MEASure:FUNCtion<x>[:STATe]

Sets or queries the on/off status of the specified user-defined function.

Syntax
```
:MEASure:FUNCtion<x>[:
STATe] {<Boolean>}
:MEASure:FUNCtion<x>:STATe?
```
<x> = 1 to 20 (F1 to F20)

Example
```
:MEASURE:FUNCTION1:STATE ON
:MEASURE:FUNCTION1:STATE?
-> :MEASURE:FUNCTION1:STATE 1
```

:MEASure:FUNCtion<x>:UNIT

Sets or queries the unit that is added to the computation result of the specified user-defined function.

Syntax
```
:MEASure:FUNCtion<x>:UNIT {<String>}
:MEASure:FUNCtion<x>:UNIT?
```
<x> = 1 to 20 (F1 to F20)
<String> = Up to 8 characters

Example
```
:MEASURE:FUNCTION1:UNIT "W"
:MEASURE:FUNCTION1:UNIT?
-> :MEASURE:FUNCTION1:UNIT "W"
```

Description This command has no effect on the computation result.

:MEASure:MHOLd

Sets or queries the on/off status of the MAX HOLD feature used in user-defined functions.

Syntax
```
:MEASure:MHOLd {<Boolean>}
:MEASure:MHOLd?
```
Example
```
:MEASURE:MHOLD ON
:MEASURE:MHOLD?
-> :MEASURE:MHOLD 1
```
Description
- The MAX HOLD operation starts when the MAX HOLD feature is specified by a user-defined function and :MEASure:MHOLd is set to ON.
- When :MEASure:MHOLd is set to OFF, the MAX HOLD operation ends, and the MAX HOLD value becomes “no data.”
- If :MEASure:MHOLd is set to ON after having already been set to ON before, the MAX HOLD value is reset, and the MAX HOLD operation starts again.
- For information about specifying the MAX HOLD feature, see the features guide, IM WT1801R-01EN.

:MEASure:PC?

Queries all Pc (corrected power) computation settings.

Syntax
```
:MEASure:PC?
```

:MEASure:PC:IEC

Sets or queries the Pc (corrected power) equation.

Syntax
```
:MEASure:PC:IEC {<NRf>}
:MEASure:PC:IEC?
```
<NRf> = 1976, 2011

Example
```
:MEASURE:PC:IEC 1976
:MEASURE:PC:IEC?
-> :MEASURE:PC:IEC 1976
```
Description
- Specify the publication year of the IEC76-1 in which the Pc equation that you want to use was written.
- If 2011 is specified, “1993 (publication year of the previous edition)” is returned as a response to a setting query.

:MEASure:PC:P<x>

Sets or queries a Pc (corrected power) equation parameter.

Syntax
```
:MEASure:PC:P<x> {<NRf>}
:MEASure:PC:P<x>?
```
<x> = 1, 2 (P1, P2)

<NRf> = 0.0001 to 9.9999

Example
```
:MEASURE:PC:P1 0.5
:MEASURE:PC:P1? -> :MEASURE:PC:
P1 0.5000
```
Description These parameters are used when :MEASure:PC:IEC is set to 1976 (IEC76-1 1976).

:MEASure:PHASe

Sets or queries the display format of the phase difference.

Syntax
```
:MEASure:PHASe {<NRf>}
:MEASure:PHASe?
```
<NRf> = 180, 360

Example
```
:MEASURE:PHASE 180
:MEASURE:PHASE?
-> :MEASURE:PHASE 180
```
Description When 180 is selected, the phase is displayed using 0 to ±180° (Lead/Lag). When 360 is selected, the phase is displayed using 0° to 360°.

:MEASure:SAMPling

Sets or queries the sampling frequency.

Syntax
```
:MEASure:SAMPling {AUTO|CLKA|
CLKB|CLKC}
:MEASure:SAMPling?
```
Example
```
:MEASURE:SAMPLING AUTO
:MEASURE:SAMPLING?
-> :MEASURE:SAMPLING AUTO
```
Description For details on the available sampling frequencies, see the features guide, IM WT1801R-01EN.

:MEASure:SFORmula

Sets or queries the equation used to compute S (apparent power).

Syntax
```
:MEASure:SFORmula {RMS|MEAN|DC|
MRMS|RMEAN}
:MEASure:SFORmula?
```
Example
```
:MEASURE:SFORMULA RMS
:MEASURE:SFORMULA?
-> :MEASURE:SFORMULA RMS
```
Description The equations that correspond to each option are as follows:

RMS: S = Urms*Irms
MEAN: S = Umean*Imean
DC: S = Udc*Idc
MRMS: S = Umean*Irms
RMEAN: S = Urmean*Irmean

:MEASure:SQFormula

Sets or queries the equation used to compute S (apparent power) and Q (reactive power).

Syntax
```
:MEASure:SQFormula {TYPE1|TYPE2|
TYPE3}
:MEASure:SQFormula?
```
Example
```
:MEASURE:SQFORMULA TYPE1
:MEASURE:SQFORMULA?
-> :MEASURE:SQFORMULA TYPE1
```
Description
- For details on the equations available for TYPE1, TYPE2, or TYPE3, see the features guide, IM WT1801R-01EN.
- TYPE3 can only be selected on models with the harmonic measurement (/G5 or /G6) option.

:MEASure:SYNChronize

Sets or queries the synchronized measurement mode.

Syntax
```
:MEASure:SYNChronize {MASTer|SLAVe}
:MEASure:SYNChronize?
```

Example
```
:MEASURE:SYNCHRONIZE MASTER
:MEASURE:SYNCHRONIZE?
-> :MEASURE:SYNCHRONIZE MASTER
```

5.16 MOTor Group

The commands in this group deal with the motor evaluation function.
You can make the same settings and queries that you can make by pressing MOTOR/AUX SET (SHIFT+SCALING) on the front panel.
The commands in this group are only valid on models with the motor evaluation function (/MTR) option.

:MOTor?

Queries all motor evaluation function settings.

Syntax :MOTor?

:MOTor:EANGle?

Queries all electrical angle measurement settings.

Syntax :MOTor:EANGle?

Description The electrical angle measurement feature is only available on models with the harmonic measurement (/G5 or /G6) option.

:MOTor:EANGle:CORRection?

Queries all electrical angle correction settings.

Syntax :MOTor:EANGle:CORRection?

:MOTor:EANGle:CORRection:AENTer?

Queries all automatic electrical angle correction entry settings.

Syntax :MOTor:EANGle:CORRection:AENTer?

:MOTor:EANGle:CORRection:AENTer[:EXECute]

Executes an automatic electrical angle correction entry.

Syntax :MOTor:EANGle:CORRection:AENTer[:EXECute]

Example :MOTOR:EANGLE:CORRECTION:AENTER:EXECUTE

Description The measured electrical angle of the target source (:MOTor:EANGle:CORRection:AENTer:TARGet) is entered as the correction value.

:MOTor:EANGle:CORRection:AENTer:TARGet

Sets or queries the target source for automatically entering the electrical angle correction value.

Syntax :MOTor:EANGle:CORRection:AENTer:TARGet {U<x>|I<x>}
:MOTor:EANGle:CORRection:AENTer:TARGet?
<x> = 1 to 6 (element)

Example :MOTOR:EANGLE:CORRECTION:AENTER:TARGET U1
:MOTOR:EANGLE:CORRECTION:AENTER:TARGET?
-> :MOTOR:EANGLE:CORRECTION:AENTER:TARGET U1

:MOTor:EANGle:CORRection:CLEar

Clears the electrical angle correction value.

Syntax :MOTor:EANGle:CORRection:CLEar

Example :MOTOR:EANGLE:CORRECTION:CLEAR

:MOTor:EANGle:CORRection[:VALue]

Sets or queries the electrical angle correction value.

Syntax :MOTor:EANGle:CORRection[:VALue] {<NRf>}
:MOTor:EANGle:CORRection:VALue?
<NRf> = -180.00 to 180.00

Example :MOTOR:EANGLE:CORRECTION:VALUE 0
:MOTOR:EANGLE:CORRECTION:VALUE?
-> :MOTOR:EANGLE:CORRECTION:VALUE 0.00

:MOTor:EANGle[:STATe]

Sets or queries the on/off status of electrical angle measurement.

Syntax :MOTor:EANGle[:STATe] {<Boolean>}
:MOTor:EANGle:STATe?

Example :MOTOR:EANGLE:STATE ON
:MOTOR:EANGLE:STATE?
-> :MOTOR:EANGLE:STATE 1

:MOTor:FILTer?

Queries all input filter settings.

Syntax :MOTor:FILTer?

:MOTor:FILTer[:LINE]

Sets or queries the line filter.

Syntax :MOTor:FILTer[:LINE] {OFF|<Frequency>}
:MOTor:FILTer:LINE?
OFF = Line filter off
<Frequency> = 100 Hz, 1 kHz (line filter on; cutoff frequency)

Example :MOTOR:FILTER:LINE OFF
:MOTOR:FILTER:LINE?
-> :MOTOR:FILTER:LINE OFF

:MOTor:PM?

Queries all motor output (Pm) settings.

Syntax :MOTor:PM?

:MOTor:PM:SCALing

Sets or queries the motor output computation scaling factor.

Syntax :MOTor:PM:SCALing {<NRf>}
:MOTor:PM:SCALing?
<NRf> = 0.0001 to 99999.9999

Example :MOTOR:PM:SCALING 1
:MOTOR:PM:SCALING?
-> :MOTOR:PM:SCALING 1.0000

:MOTor:PM:UNIT

Sets or queries the unit that is added to the motor output computation result.

Syntax :MOTor:PM:UNIT {<String>}
:MOTor:PM:UNIT?
<String> = Up to 8 characters

Example :MOTOR:PM:UNIT "W"
:MOTOR:PM:UNIT? -> :MOTOR:PM:UNIT "W"

Description This command has no effect on the computation result.

:MOTor:POLE

Sets or queries the motor's number of poles.

Syntax :MOTor:POLE {<NRf>}
:MOTor:POLE?
<NRf> = 1 to 99

Example :MOTOR:POLE 2
:MOTOR:POLE? -> :MOTOR:POLE 2

:MOTor:SPEed?

Queries all rotating speed (Speed) settings.

Syntax :MOTor:SPEed?

:MOTor:SPEed:AUTO

Sets or queries the voltage auto range on/off status of the revolution signal (analog input type).

Syntax :MOTor:SPEed:AUTO {<Boolean>}
:MOTor:SPEed:AUTO?

Example :MOTOR:SPEED:AUTO ON
:MOTOR:SPEED:AUTO?
-> :MOTOR:SPEED:AUTO 1

Description This command is valid when the revolution signal input type (:MOTor:SPEed:TYPE) is set to ANALog (analog input).

:MOTor:SPEed:LSCale?

Queries all revolution signal (analog input type) linear scaling settings.

Syntax :MOTor:SPEed:LSCale?

Description Linear scaling settings are valid when the revolution signal input type (:MOTor:SPEed:TYPE) is set to ANALog (analog input).

:MOTor:SPEed:LSCale:AVALue

Sets or queries the slope (A) of the linear scale of the revolution signal (analog input type).

Syntax :MOTor:SPEed:LSCale:AVALue {<NRf>}
:MOTor:SPEed:LSCale:AVALue?
<NRf> = 1.000E-03 to 1.000E+06

Example :MOTOR:SPEED:LSCALE:AVALUE 1.000
:MOTOR:SPEED:LSCALE:AVALUE?
-> :MOTOR:SPEED:LSCALE:
AVALUE 1.000E+00

:MOTor:SPEed:LSCale:BVALue

Sets or queries the offset (B) of the linear scale of the revolution signal (analog input type).

Syntax :MOTor:SPEed:LSCale:BVALue {<NRf>}
:MOTor:SPEed:LSCale:BVALue?
<NRf> = -1.000E+06 to 1.000E+06

Example :MOTOR:SPEED:LSCALE:BVALUE 0
:MOTOR:SPEED:LSCALE:BVALUE?
-> :MOTOR:SPEED:LSCALE:
BVALUE 0.000E+00

:MOTor:SPEed:LSCale:CALCulate?

Queries all parameter calculation settings for the linear scale of the revolution signal (analog input type).

Syntax :MOTor:SPEed:LSCale:CALCulate?

:MOTor:SPEed:LSCale:CALCulate:{P1X|P1Y|P2X|P2Y}

Sets or queries the data (Point1X, Point1Y, Point2X, or Point2Y) for parameter calculations of the linear scale of the revolution signal (analog input type).

Syntax :MOTor:SPEed:LSCale:CALCulate:
{P1X|P1Y|P2X|P2Y} {<NRf>}
:MOTor:SPEed:LSCale:CALCulate:
{P1X|P1Y|P2X|P2Y}?
<NRf> = -1.000E+12 to 1.000E+12

Example :MOTOR:SPEED:LSCALE:CALCULATE:P1X 0
:MOTOR:SPEED:LSCALE:CALCULATE:P1X?
-> :MOTOR:SPEED:LSCALE:CALCULATE:
P1X 0.000E+00

:MOTor:SPEed:LSCale:CALCulate:EXECute

Calculates parameters for the linear scale of the revolution signal (analog input type).

Syntax :MOTor:SPEed:LSCale:CALCulate:
EXECute

Example :MOTOR:SPEED:LSCALE:CALCULATE:
EXECUTE

Description This instrument uses the data that has been specified (Point1X, Point1Y, Point2X, and Point2Y) to calculate and set the slope (A) and offset (B) of the linear scale.

:MOTor:SPEed:PRANge

Sets or queries the range of the revolution signal (pulse input type).

Syntax :MOTor:SPEed:PRANge {<NRf>,<NRf>}
:MOTor:SPEed:PRANge?
<NRf> = 0.0000 to 99999.9999

Example :MOTOR:SPEED:PRANGE 10000,0
:MOTOR:SPEED:PRANGE?
-> :MOTOR:SPEED:PRANGE 10000.0000,
0.0000

Description
- Set the upper limit and then the lower limit.
- This command is valid when the revolution signal input type (:MOTor:SPEed:TYPE) is set to PULSe (pulse input).

:MOTor:SPEed:PULSe

Sets or queries the number of pulses of the revolution signal (pulse input type).

Syntax :MOTor:SPEed:PULSe {<NRf>}
:MOTor:SPEed:PULSe?
<NRf> = 1 to 9999

Example :MOTOR:SPEED:PULSE 60
:MOTOR:SPEED:PULSE?
-> :MOTOR:SPEED:PULSE 60

Description This command is valid when the revolution signal input type (:MOTor:SPEed:TYPE) is set to PULSe (pulse input).

:MOTor:SPEed:RANGe

Sets or queries the voltage range of the revolution signal (analog input type).

Syntax :MOTor:SPEed:RANGe {<Voltage>}
:MOTor:SPEed:RANGe?
<Voltage> = 1 V, 2 V, 5 V, 10 V, 20 V

Example :MOTOR:SPEED:RANGE 20V
:MOTOR:SPEED:RANGE?
-> :MOTOR:SPEED:RANGE 20.0E+00

Description This command is valid when the revolution signal input type (:MOTor:SPEed:TYPE) is set to ANALog (analog input).

:MOTor:SPEed:SCALing

Sets or queries the rotating speed computation scaling factor.

Syntax :MOTor:SPEed:SCALing {<NRf>}
:MOTor:SPEed:SCALing?
<NRf> = 0.0001 to 99999.9999

Example :MOTOR:SPEED:SCALING 1
:MOTOR:SPEED:SCALING?
-> :MOTOR:SPEED:SCALING 1.0000

:MOTor:SPEed:TYPE

Sets or queries the revolution signal input type.

Syntax :MOTor:SPEed:TYPE {ANALog|PULSe}
:MOTor:SPEed:TYPE?

Example :MOTOR:SPEED:TYPE ANALOG
:MOTOR:SPEED:TYPE?
-> :MOTOR:SPEED:TYPE ANALOG

:MOTor:SPEed:UNIT

Sets or queries the unit that is added to the rotating speed computation result.

Syntax :MOTor:SPEed:UNIT {<String>}
:MOTor:SPEed:UNIT?
<String> = Up to 8 characters

Example :MOTOR:SPEED:UNIT "rpm"
:MOTOR:SPEED:UNIT?
-> :MOTOR:SPEED:UNIT "rpm"

Description This command has no effect on the computation result.

:MOTor:SSPeed

Sets or queries the frequency measurement source for the synchronous speed (SyncSp) computation.

Syntax :MOTor:SSPeed {U<x>|I<x>}
:MOTor:SSPeed?
<x> = 1 to 6 (element)

Example :MOTOR:SSPEED I1
:MOTOR:SSPEED? -> :MOTOR:SSPEED I1

:MOTor:SYNChronize

Sets or queries the synchronization source for the rotating speed (Speed) and torque (Torque) computations.

Syntax :MOTor:SYNChronize {U<x>|I<x>|
EXTernal|NONE}
:MOTor:SYNChronize?
<x> = 1 to 6 (element)
EXTernal = External clock input (Ext Clk)
NONE = No synchronization source

Example :MOTOR:SYNCHRONIZE NONE
:MOTOR:SYNCHRONIZE?
-> :MOTOR:SYNCHRONIZE NONE

:MOTor:TORQue?

Queries all torque (Torque) settings.

Syntax :MOTor:TORQue?

:MOTor:TORQue:AUTO

Sets or queries the voltage auto range on/off status of the torque signal (analog input type).

Syntax
```
:MOTor:TORQue:AUTO {<Boolean>}
:MOTor:TORQue:AUTO?
```
Example
```
:MOTOR:TORQUE:AUTO ON
:MOTOR:TORQUE:AUTO?
-> :MOTOR:TORQUE:AUTO 1
```
Description This command is valid when the torque signal input type (:MOTor:TORQue:TYPE) is set to ANALog (analog input).

:MOTor:TORQue:LSCale?

Queries all torque signal (analog input type) linear scaling settings.

Syntax
```
:MOTor:TORQue:LSCale?
```
Description Linear scaling settings are valid when the torque signal input type (:MOTor:TORQue:TYPE) is set to ANALog (analog input).

:MOTor:TORQue:LSCale:AVALue

Sets or queries the slope (A) of the linear scale of the torque signal (analog input type).

Syntax
```
:MOTor:TORQue:LSCale:AVALue {<NRf>}
:MOTor:TORQue:LSCale:AVALue?
```
<NRf> = 1.000E-03 to 1.000E+06

Example
```
:MOTOR:TORQUE:LSCALE:AVALUE 1.000
:MOTOR:TORQUE:LSCALE:AVALUE?
-> :MOTOR:TORQUE:LSCALE:
AVALUE 1.000E+00
```

:MOTor:TORQue:LSCale:BVALue

Sets or queries the offset (B) of the linear scale of the torque signal (analog input type).

Syntax
```
:MOTor:TORQue:LSCale:BVALue {<NRf>}
:MOTor:TORQue:LSCale:BVALue?
```
<NRf> = -1.000E+06 to 1.000E+06

Example
```
:MOTOR:TORQUE:LSCALE:BVALUE 0
:MOTOR:TORQUE:LSCALE:BVALUE?
-> :MOTOR:TORQUE:LSCALE:
BVALUE 0.000E+00
```

:MOTor:TORQue:LSCale:CALCulate?

Queries all parameter calculation settings for the linear scale of the torque signal (analog input type).

Syntax
```
:MOTor:TORQue:LSCale:CALCulate?
```

:MOTor:TORQue:LSCale:CALCulate:{P1X|P1Y|P2X|P2Y}

Sets or queries the data (Point1X, Point1Y, Point2X, or Point2Y) for parameter calculations of the linear scale of the torque signal (analog input type).

Syntax
```
:MOTor:TORQue:LSCale:CALCulate:
{P1X|P1Y|P2X|P2Y} {<NRf>}
:MOTor:TORQue:LSCale:CALCulate:
{P1X|P1Y|P2X|P2Y}?
```
<NRf> = -1.000E+12 to 1.000E+12

Example
```
:MOTOR:TORQUE:LSCALE:CALCULATE:P1X 0
:MOTOR:TORQUE:LSCALE:CALCULATE:P1X?
-> :MOTOR:TORQUE:LSCALE:CALCULATE:
P1X 0.000E+00
```

:MOTor:TORQue:LSCale:CALCulate:EXECute

Calculates parameters for the linear scale of the torque signal (analog input type).

Syntax
```
:MOTor:TORQue:LSCale:CALCulate:
EXECute
```
Example
```
:MOTOR:TORQUE:LSCALE:CALCULATE:
EXECUTE
```
Description This instrument uses the data that has been specified (Point1X, Point1Y, Point2X, and Point2Y) to calculate and set the slope (A) and offset (B) of the linear scale.

:MOTor:TORQue:PRANge

Sets or queries the range of the torque signal (pulse input type).

Syntax
```
:MOTor:TORQue:PRANge {<NRf>,<NRf>}
:MOTor:TORQue:PRANge?
```
<NRf> = -10000.0000 to 10000.0000

Example
```
:MOTOR:TORQUE:PRANGE 50,-50
:MOTOR:TORQUE:PRANGE?
-> :MOTOR:TORQUE:PRANGE 50.0000,
-50.0000
```
Description
- Set the upper limit and then the lower limit.
- This command is valid when the torque signal input type (:MOTor:TORQue:TYPE) is set to PULSe (pulse input).

:MOTor:TORQue:RANGe

Sets or queries the voltage range of the torque signal (analog input type).

Syntax
```
:MOTor:TORQue:RANGe {<Voltage>}
:MOTor:TORQue:RANGe?
```
<Voltage> = 1 V, 2 V, 5 V, 10 V, 20 V

Example
```
:MOTOR:TORQUE:RANGE 20V
:MOTOR:TORQUE:RANGE?
-> :MOTOR:TORQUE:RANGE 20.0E+00
```
Description This command is valid when the torque signal input type (:MOTor:TORQue:TYPE) is set to ANALog (analog input).

:MOTor:TORQue:RATE?

Queries all torque signal (pulse input type) rated-value settings.

Syntax :MOTor:TORQue:RATE?

:MOTor:TORQue:RATE:{UPPer|LOWer}

Sets or queries the upper or lower limit of the rated value of the torque signal (pulse input type).

Syntax :MOTor:TORQue:RATE:{UPPer|
LOWer} {<NRf>,<Frequency>}
:MOTor:TORQue:RATE:{UPPer|LOWer}?
<NRf> = -10000.0000 to 10000.0000
<Frequency> = 1 Hz to 100 MHz

Example :MOTOR:TORQUE:RATE:UPPER 50,15KHZ
:MOTOR:TORQUE:RATE:UPPER?
-> :MOTOR:TORQUE:RATE:UPPER 50.0000,
15.000E+03

Description This command is valid when the torque signal input type (:MOTor:TORQue:TYPE) is set to PULSe (pulse input).

:MOTor:TORQue:SCALing

Sets or queries the torque computation scaling factor.

Syntax :MOTor:TORQue:SCALing {<NRf>}
:MOTor:TORQue:SCALing?
<NRf> = 0.0001 to 99999.9999

Example :MOTOR:TORQUE:SCALING 1
:MOTOR:TORQUE:SCALING?
-> :MOTOR:TORQUE:SCALING 1.0000

:MOTor:TORQue:TYPE

Sets or queries the torque signal input type.

Syntax :MOTor:TORQue:TYPE {ANALog|PULSe}
:MOTor:TORQue:TYPE?

Example :MOTOR:TORQUE:TYPE ANALOG
:MOTOR:TORQUE:TYPE?
-> :MOTOR:TORQUE:TYPE ANALOG

:MOTor:TORQue:UNIT

Sets or queries the unit that is added to the torque computation result.

Syntax :MOTor:TORQue:UNIT {<String>}
:MOTor:TORQue:UNIT?
<String> = Up to 8 characters

Example :MOTOR:TORQUE:UNIT "Nm"
:MOTOR:TORQUE:UNIT?
-> :MOTOR:TORQUE:UNIT "Nm"

Description This command has no effect on the computation result.

5.17 NUMeric Group

The command in this group deal with numeric data output.
There are no front panel keys that correspond to the commands in this group.

:NUMeric?

Queries all numeric data output settings.

Syntax :NUMeric?

:NUMeric:BYTeorder

Sets or queries the output byte order of the numeric data (FLOAT format).

Syntax :NUMeric:BYTeorder {LSBFirst|MSBFirst}
:NUMeric:BYTeorder?

Example :NUMERIC:BYTEORDER LSBFIRST
:NUMERIC:BYTEORDER?
-> :NUMERIC:BYTEORDER LSBFIRST

Description
- This command is valid when :NUMeric:FORMat is set to FLOat.
- The default setting on the WT1800R is LSBFirst. If you specified a command type that is compatible with legacy models (see chapter 8), the default setting is MSBFirst.

:NUMeric:FORMat

Sets or queries the numeric data format.

Syntax :NUMeric:FORMat {ASCii|FLOat}
:NUMeric:FORMat?

Example :NUMERIC:FORMAT ASCII
:NUMERIC:FORMAT?
-> :NUMERIC:FORMAT ASCII

Description
- The format of the numeric data that is output varies depending on how this command is set. The different formats are explained below.
  - When the format is set to ASCii:
    Physical values are output in <NR3> format. (Only the elapsed integration time—TIME—is output in <NR1> format.)
    The data items are separated by commas.
  - When the format is set to FLOat:
    A header (for example, “#260” or “#3208”) is added in front of each numeric data block.
    A physical value in IEEE single-precision floating point (4-byte) format follows the header.
    The output byte order of the data of each item follows the order that is set by using the “:NUMeric:BYTeorder” command.
- For the formats of each individual numeric data item, see “Numeric data formats” on page 5-97 at the end of this group of commands.

:NUMeric:HSPeed?

Queries all numeric data output settings of the high speed data capturing mode.

Syntax :NUMeric:HSPEED?

Description The number of numeric data items output by :NUMeric:HSPeed:ITEM<x> is determined by :NUMeric:HSPeed:NUMber.

:NUMeric:HSPeed:CLEar

Clears high speed data capturing mode numeric list data output items (sets the items to NONE).

Syntax :NUMeric:HSPeed:CLEar {ALL|<NRf>[,<NRf>]}
ALL = Clear all items
First <NRf> = 1 to 30 (the number of the first item to clear)
Second <NRf> = 1 to 30 (the number of the last item to clear)

Example :NUMERIC:HSPEED:CLEAR ALL

Description If the second <NRf> is omitted, the output item specified by the first <NRf> and all following output items (up to number 30) are cleared.

:NUMeric:HSPeed:DELete

Deletes high speed data capturing mode numeric list data output items.

Syntax :NUMeric:HSPeed:DELete {<NRf>[,<NRf>]}
First <NRf> = 1 to 30 (the number of the first item to delete)
Second <NRf> = 1 to 30 (the number of the last item to delete)

Example :NUMERIC:HSPEED:DELETE 1
(Deletes ITEM1 and shifts ITEM2 and subsequent items forward)
:NUMERIC:HSPEED:DELETE 1,3
(Deletes ITEM1 to ITEM3 and shifts ITEM4 and subsequent items forward)

Description
- The positions of deleted output items are filled by the items that follow them, and empty sections at the end are set to NONE.
- If the second <NRf> is omitted, only the output item specified by the first <NRf> is deleted.

:NUMeric:HSPeed:HEADer?

Queries the header of the numeric data of high speed data capturing mode.

Syntax `:NUMeric:HSPeed:HEADer? {<NRf>}`
<NRf> = 1 to 30 (item number)

Example
- When <NRf> is specified:
  `:NUMERIC:HSPEED:HEADER? 1 -> U-E1`
- When <NRf> is omitted (when :NUMeric:HSPeed:NUMber is set to 3):
  `:NUMERIC:HSPEED:HEADER?`
  `-> U-E1,I-E1,P-E1`

Description
- The data name (header) of the output item is generated.
- If <NRf> is specified, only the data name for the specified item number is output.
- If <NRf> is omitted, the data names of the items from 1 to the number specified by the :NUMeric:HSPeed:NUMber command are output in order.

:NUMeric:HSPeed:ITEM<x>

Sets or queries the output item (function and element) of the specified high speed data capturing mode numeric data item.

Syntax `:NUMeric:HSPeed:ITEM<x> {NONE|<Function>[,<Element>]}`
`:NUMeric:HSPeed:ITEM<x>?`
<x> = 1 to 30 (item number)
NONE = No output item
<Function> = {U|I|P|SPEed|TORQue|PM|AUX<x>}
(<x> = 1 to 2)
<Element> = {<NRf>|SIGMa|SIGMB|SIGMC}
(<NRf> = 1 to 6)

Example `:NUMERIC:HSPEED:ITEM1 U,1`
`:NUMERIC:HSPEED:ITEM1?`
`-> :NUMERIC:HSPEED:ITEM1 U,1`

Description
- If <Element> is omitted, the element is set to 1.
- <Element> is omitted from responses to functions that do not need it.
- SPEed, TORQue and PM are only valid on models with the motor evaluation function (/MTR) option.
- AUX<x> is only valid on models with the auxiliary input (/AUX) option.

:NUMeric:HSPeed:{MAXimum|MINimum}?

Queries the maximum or minimum value of the numeric data of high speed data capturing mode.

Syntax `:NUMeric:HSPeed:{MAXimum|MINimum}? {<NRf>}`
<NRf> = 1 to 30 (item number)

Example
- When <NRf> is specified:
  `:NUMERIC:HSPEED:MAXIMUM? 1`
  `-> 103.79E+00`
- When <NRf> is omitted (when :NUMeric :HSPeed:NUMber is set to 3):
  `:NUMERIC:HSPEED:MAXIMUM?`
  `-> 103.79E+00,1.0185E+00,105.27E+00`
- When :NUMeric:FORMat is set to FLOat:
  `:NUMERIC:HSPEED:MAXIMUM?`
  `-> #N` (N-digit byte number; data byte sequence)

Description
- The maximum value or minimum value of all the numeric data from the time that capturing started to the current data update is output.
- If <NRf> is specified, only the maximum value or minimum value of the numeric data of the specified item number is output.
- If <NRf> is omitted, the maximum values or minimum values of the numeric data of the items from 1 to the number specified by the :NUMeric:HSPeed:NUMber command are output in order.

:NUMeric:HSPeed:NUMber

Sets or queries the number of numeric data items that are transmitted by the :NUMeric:HSPeed:VALue? command.

Syntax `:NUMeric:HSPeed:NUMber {<NRf>}`
`:NUMeric:HSPeed:NUMber?`
<NRf> = 1 to 30

Example `:NUMERIC:HSPEED:NUMBER 3`
`:NUMERIC:HSPEED:NUMBER`
`-> :NUMERIC:HSPEED:NUMBER 3`

Description
- If you omit the parameters to the :NUMeric:HSPeed:VALue? query, as many data records as are captured in a single data update interval are output in order. A single data record contains the numeric data items from number 1 to the specified value.
- By default, the number of numeric data items is set to 3.

:NUMeric:HSPeed:PRESet

Presets the numeric data output item pattern of the high speed data capturing mode.

Syntax :NUMeric:HSPeed:PRESet {DEFault|RECord}
DEFault = default
RECord = same settings as the file output

Example :NUMERIC:HSPEED:PRESET DEFAULT

Description For the default, see “Preset patterns for high speed data capturing mode numeric data output items” on page 5-101 at the end of the commands for this group.

:NUMeric:HSPeed:VALue?

Queries the numeric data of high speed data capturing mode.

Syntax :NUMeric:HSPeed:VALue? {<NRf>}
<NRf> = 1 to 30 (item number)

Example When there are 100 data records captured in a single data update interval:

- When <NRf> is specified:
  :NUMERIC:HSPEED:VALUE? 1
  -> 103.79E+00,103.26E+00, ... (omitted)
  ...,103.53E+00(U1[1],U1[2], ... (omitted)
  ..., U1[100]:100 data entries)
- When <NRf> is omitted (when :NUMeric:HSPeed:NUMber is set to 3):
  :NUMERIC:HSPEED:VALUE?
  -> 103.79E+00,1.0143E+00,105.27E+00,
  103.26E+00,1.0185E+00,105.17E+00,
  ... (omitted) ...,103.53E+00,1.0164E+00,
  105.23E+00(U1[1],I1[1],P1[1],U1[2],
  I1[2],P1[2], ... (omitted) ...,U1[100],
  I1[100],P1[100]:300 = 3 × 100 data entries)
- When:NUMeric:FORMat is set to FLOat:
  :NUMERIC:HSPEED:VALUE?
  -> #N (N-digit byte number; data byte sequence)

Description
- The number of data records that were captured in a single data update interval are output in the order that they were captured.
- A single numeric data record contains the following numeric data.
  - If <NRf> is omitted, the numeric data record contains the numeric data of the items from 1 to the number specified by the :NUMeric:HSPeed:NUMber command (up to 30 items).
  - If <NRf> is specified, the numeric data record contains the single numeric data entry of the specified item number, regardless of the value set by the :NUMeric:HSPeed:NUMber command.
- In ASCII format, numeric data entries and data records are both separated by commas.
- For the formats of the individual numeric data items that are output, see “Numeric data formats” on page 5-97 at the end of this group of commands.
- Before data capturing starts or after measurement conditions are changed, the number of data captures is “0.” Therefore, there are no data records. The response that is returned is described below.
  - In ASCII format, there is no response. Only “<RMT>“ is returned.
  - In FLOAT format, the response is “#10” (only the header that indicates there are no data bytes is returned).

:NUMeric:HOLD

Sets or queries the on/off (hold/release) status of the numeric data hold feature.

Syntax
```
:NUMeric:HOLD {<Boolean>}
:NUMeric:HOLD?
```
Example
```
:NUMERIC:HOLD ON
:NUMERIC:HOLD? -> :NUMERIC:HOLD 1
```
Description
- If :NUMeric:HOLD is set to ON before :NUMeric[:NORMal]:VALue? or :NUMeric:LIST:VALue? is executed, all the numeric data at that point in time can be held internally.
- As long as :NUMeric:HOLD is set to ON, numeric data is held even when the numeric data on the screen is updated.
  For example, if you wish to retrieve various types of numeric data from each element at the same point in time, use the following commands:
  ```
  :NUMeric:HOLD ON
  :NUMeric[:NORMal]:
  ITEM1 URMS,1;ITEM2 IRMS,1;...
  ```
  (Set the numeric data items of element 1.)
  ```
  :NUMeric[:NORMal]:VALue?
  ```
  (Receive the numeric data of element 1.)
  ```
  :NUMeric[:NORMal]:
  ITEM1 URMS,2;ITEM2 IRMS,2;...
  ```
  (Set the numeric data items of element 2.)
  ```
  :NUMeric[:NORMal]:VALue?
  ```
  (Receive the numeric data of element 2.)
  ```
  :NUMeric[:NORMal]:
  ITEM1 URMS,3;ITEM2 IRMS,3;...
  ```
  (Set the numeric data items of element 3.)
  ```
  :NUMeric[:NORMal]:VALue?
  ```
  (Receive the numeric data of element 3.)
  ```
  :NUMeric:HOLD OFF
  ```
- If :NUMeric:HOLD is set to ON after having already been set to ON before, the numeric data is cleared, and the most recent numeric data is held internally. When retrieving numeric data continuously, this method can be used to circumvent the need to repeatedly set :NUMeric:HOLD to OFF.

:NUMeric:LIST?

Queries all harmonic measurement numeric list data output settings.

Syntax
```
:NUMeric:LIST?
```
Description
- This is only valid on models with the harmonic measurement (/G5 or /G6) option.
- The number of numeric list data items output by :NUMeric:LIST:ITEM<x> is determined by :NUMeric:LIST:NUMber.

:NUMeric:LIST:CLEar

Clears harmonic measurement numeric list data output items (sets the items to NONE).

Syntax
```
:NUMeric:LIST:CLEar {ALL|<NRf>
[,<NRf>]}
```
ALL = Clear all items
First <NRf> = 1 to 64 (the number of the first item to clear)
Second <NRf> = 1 to 64 (the number of the last item to clear)

Example
```
:NUMERIC:LIST:CLEAR ALL
```
Description
- This is only valid on models with the harmonic measurement (/G5 or /G6) option.
- If the second <NRf> is omitted, the output item specified by the first <NRf> and all following output items (up to number 64) are cleared.

:NUMeric:LIST:DELete

Deletes harmonic measurement numeric list data output items.

Syntax
```
:NUMeric:LIST:DELete {<NRf>[,<NRf>]}
```
First <NRf> = 1 to 64 (the number of the first item to delete)
Second <NRf> = 1 to 64 (the number of the last item to delete)

Example
```
:NUMERIC:LIST:DELETE 1
```
(Deletes ITEM1 and shifts ITEM2 and subsequent items forward)
```
:NUMERIC:LIST:DELETE 1,3
```
(Deletes ITEM1 to ITEM3 and shifts ITEM4 and subsequent items forward)

Description
- This is only valid on models with the harmonic measurement (/G5 or /G6) option.
- The positions of deleted output items are filled by the items that follow them, and empty sections at the end are set to NONE.
- If the second <NRf> is omitted, only the output item specified by the first <NRf> is deleted.

:NUMeric:LIST:ITEM<x>

Sets or queries the output item (function and element) of the specified harmonic measurement numeric list data item.

Syntax
```
:NUMeric:LIST:ITEM<x>{NONE|
<Function>,<Element>}
:NUMeric:LIST:ITEM<x>?
```
<x> = 1 to 64 (item number)
NONE = No output item
<Function> = {U|I|P|S|Q|LAMBda|PHI|PHIU|PHII|Z|RS|XS|RP|XP|UHDF|IHDF|PHDF}
<Element> = {<NRf>|SIGMa|SIGMB|SIGMC} (<NRf> = 1 to 6)

Example
```
:NUMERIC:LIST:ITEM1 U,1
:NUMERIC:LIST:ITEM1?
-> :NUMERIC:LIST:ITEM1 U,1
```

Description
- This is only valid on models with the harmonic measurement (/G5 or /G6) option.
- For information about the options available for <Function>, see “Numeric list data functions” in “Function option list” on page 5-44.

:NUMeric:LIST:NUMber

Sets or queries the number of numeric list data items that are transmitted by :NUMeric:LIST:VALue?.

Syntax
```
:NUMeric:LIST:NUMber {<NRf>|ALL}
:NUMeric:LIST:NUMber?
```
<NRf> = 1 to 64 (ALL)

Example
```
:NUMERIC:LIST:NUMBER 5
:NUMERIC:LIST:NUMBER
-> :NUMERIC:LIST:NUMBER 5
```

Description
- This is only valid on models with the harmonic measurement (/G5 or /G6) option.
- If the parameter is omitted from the :NUMeric:LIST:VALue? command, the numeric list data items from 1 to the specified value are output in order.
- By default, the number of numeric data items is set to 1.

:NUMeric:LIST:ORDer

Sets or queries the maximum output harmonic order of the harmonic measurement numeric list data.

Syntax
```
:NUMeric:LIST:ORDer {<NRf>|ALL}
:NUMeric:LIST:ORDer?
```
<NRf> = 1 to 500 (ALL)

Example
```
:NUMERIC:LIST:ORDER 100
:NUMERIC:LIST:ORDER?
-> :NUMERIC:LIST:ORDER 100
```

Description This is only valid on models with the harmonic measurement (/G5 or /G6) option.

:NUMeric:LIST:PRESet

Presets the harmonic measurement numeric list data output item pattern.

Syntax
```
:NUMeric:LIST:PRESet {<NRf>}
```
<NRf> = 1 to 4

Example
```
:NUMERIC:LIST:PRESET 1
```

Description
- This is only valid on models with the harmonic measurement (/G5 or /G6) option.
- For information about the output items that are preset, see “Preset patterns for numeric data items” on page 5-99 at the end of the commands for this group.
- By default, the output items of Pattern 2 are selected.

:NUMeric:LIST:SELect

Sets or queries the output components of the harmonic measurement numeric list data.

Syntax
```
:NUMeric:LIST:SELect {EVEN|ODD|ALL}
:NUMeric:LIST:SELect?
```

Example
```
:NUMERIC:LIST:SELECT ALL
:NUMERIC:LIST:SELECT?
-> :NUMERIC:LIST:SELECT ALL
```

Description
- This is only valid on models with the harmonic measurement (/G5 or /G6) option.
- The available options are explained below.
  EVEN = Outputs the components of TOTal, DC, and even-order harmonics
  ODD = Outputs the components of TOTal, DC, and odd-order harmonics
  ALL = Outputs all components

:NUMeric:LIST:VALue?

Queries the harmonic measurement numeric list data.

Syntax :NUMeric:LIST:VALue? {<NRf>}
<NRf> = 1 to 64 (item number)

Example
- When <NRf> is specified:
  :NUMERIC:LIST:VALUE? 1
  -> 103.58E+00,0.00E+00,103.53E+00,
  0.09E+00,2.07E+00,0.04E+00,
  ..(omitted)..,0.01E+00,0.01E+00 (502 data items max)
- When <NRf> is omitted: (When :NUMeric:LIST:NUMber is set to 5)
  :NUMERIC:LIST:VALUE?
  -> 103.58E+00,0.00E+00,103.53E+00,
  0.09E+00,2.07E+00,0.04E+00,
  ..(omitted)..,0.00E+00,0.00E+00 (502×5 = 2510 data items max)
- When :NUMeric:FORMat is set to FLOat:
  :NUMERIC:LIST:VALUE?
  -> #N (N-digit byte number)(data byte sequence)

Description
- This is only valid on models with the harmonic measurement (/G5 or /G6) option.
- A single numeric list data item consists of up to 502 items of numeric data in the following order: TOTal, DC, 1st order, ..., :NUMeric:LIST:ORDer.
- If <NRf> is specified, only the numeric list data of the specified item number is output (up to 502 items of data).
- If <NRf> is omitted, the numeric list data of item numbers from 1 to :NUMeric:LIST:NUMber is output in order (up to 502 times the number specified by :NUMeric:LIST:NUMber).
- For the formats of the individual numeric data items that are output, see “Numeric data formats” on page 5-97 at the end of this group of commands.

:NUMeric:NORMal?

Queries all numeric data output settings.

Syntax :NUMeric:NORMal?

Description The number of numeric data items output by :NUMeric[:NORMal]:ITEM<x> is determined by :NUMeric[:NORMal]NUMber.

:NUMeric[:NORMal]:CLEar

Clears numeric data output items (sets the items to NONE).

Syntax :NUMeric[:NORMal]:CLEar {ALL|<NRf>[,<NRf>]}
ALL = Clear all items
First <NRf> = 1 to 1000 (the number of the first item to clear)
Second <NRf> = 1 to 1000 (the number of the last item to clear)

Example :NUMERIC:NORMAL:CLEAR ALL

Description If the 2nd <NRf> is omitted, the output item specified by the first <NRf> and all following output items (up to number 1000) are cleared.

:NUMeric[:NORMal]:DELete

Deletes numeric data output items.

Syntax :NUMeric[:NORMal]:DELete {<NRf>[,<NRf>]}
First <NRf> = 1 to 1000 (the number of the first item to delete)
Second <NRf> = 1 to 1000 (the number of the last item to delete)

Example :NUMERIC:NORMAL:DELETE 1
(Deletes ITEM1 and shifts ITEM2 and subsequent items forward)
:NUMERIC:NORMAL:DELETE 1,3
(Deletes ITEM1 to ITEM3 and shifts ITEM4 and subsequent items forward)

Description
- The positions of deleted output items are filled by the items that follow them, and empty sections at the end are set to NONE.
- If the second <NRf> is omitted, only the output item specified by the first <NRf> is deleted.

:NUMeric[:NORMal]:ITEM<x>

Sets or queries the specified numeric data output item (function, element, and harmonic order).

Syntax
```
:NUMeric[:NORMal]:ITEM<x> {NONE|<Function>[,<Element>][,<Order>]}
:NUMeric[:NORMal]:ITEM<x>?
```
<x> = 1 to 1000 (item number)
NONE = No output item
<Function> = {URMS|IRMS|P|S|Q|...}
<Element> = {<NRf>|SIGMa|SIGMB|SIGMC}
(<NRf> = 1 to 6)
<Order> = {TOTal|DC|<NRf>}
(<NRf> = 1 to 500)

Example
```
:NUMERIC:NORMAL:ITEM1 URMS,1
:NUMERIC:NORMAL:ITEM1?
-> :NUMERIC:NORMAL:ITEM1 URMS,1
:NUMERIC:NORMAL:ITEM1 UK,1,1
:NUMERIC:NORMAL:ITEM1?
-> :NUMERIC:NORMAL:ITEM1 UK,1,1
```

Description
- For information about the options available for <Function>, see “Numeric data functions” in “Function option list” at the end of the DISPlay group on page 5-44.
- If <Element> is omitted, the element is set to 1.
- If <Order> is omitted, the order is set to TOTal.
- <Element> and <Order> are omitted from responses to functions that do not need them.

:NUMeric[:NORMal]:NUMber

Sets or queries the number of numeric data items that are transmitted by the :NUMeric[:NORMal]:VALue? command.

Syntax
```
:NUMeric[:NORMal]:NUMber {<NRf>|ALL}
:NUMeric[:NORMal]:NUMber?
```
<NRf> = 1 to 1000 (ALL)

Example
```
:NUMERIC:NORMAL:NUMBER 15
:NUMERIC:NORMAL:NUMBER
-> :NUMERIC:NORMAL:NUMBER 15
```

Description
- If the parameter is omitted from the :NUMeric[:NORMal]:VALue? command, the numeric data items from 1 to the specified value are output in order.
- By default, the number of numeric data items is set to 15.

:NUMeric[:NORMal]:PRESet

Presets the numeric data output item pattern.

Syntax
```
:NUMeric[:NORMal]:PRESet {<NRf>}
```
<NRf> = 1 to 4

Example
```
:NUMERIC:NORMAL:PRESET 1
```

Description
- For information about the output items that are preset, see “Preset patterns for numeric data items” on page 5-99 at the end of the commands for this group.
- By default, the output items of Pattern 2 are selected.

:NUMeric[:NORMal]:VALue?

Queries the numeric data.

Syntax
```
:NUMeric[:NORMal]:VALue? {<NRf>}
```
<NRf> = 1 to 1000 (item number)

Example
- When <NRf> is specified:
```
:NUMERIC:NORMAL:VALUE? 1 ->
103.79E+00
```
- When <NRf> is omitted:
```
:NUMERIC:NORMAL:VALUE? -> 103.79E+
00,1.0143E+00,105.27E+00,..
(omitted)..,1.428E+00
```
- When :NUMeric:FORMat is set to FLOat:
```
:NUMERIC:NORMAL:VALUE? -> #N
(N-digit byte number)(data byte
sequence)
```

Description
- If <NRf> is specified, only the numeric data for the specified item is output.
- If <NRf> is omitted, the numeric data items from 1 to the number specified by the :NUMeric[:NORMal]:NUMber command are output in order.
- For the formats of the individual numeric data items that are output, see “Numeric data formats” on page 5-97 at the end of this group of commands.

Numeric data formats

Applicable commands

```
:NUMeric:FORMat                 :NUMeric:LIST:VALue?
:NUMeric:HSPeed:VALue?          :NUMeric[:NORMal]:VALue?
```

Normal data

- **The Σ of electric power values P, S, and Q**
- **Integrated values WH, WHP, WHM, AH, AHP, AHM, WS, and WQ**
- **Efficiency values ETA1, ETA2, ETA3, and ETA4; harmonic distortion factor values UHDFk, IHDFk, and PHDFk; and distortion factor values UTHD, ITHD, and PTHD.**
  ASCII: <NR3> format (mantissa: up to 6 digits, exponent: 2 digits. Example: [-]123.456E+00)
  FLOAT: IEEE single-precision floating point (4-byte) format
- **Elapsed integration time (TIME)**
  ASCII: <NR1> format in units of seconds. Example: 3600 for 1 hour (1:00:00).
  FLOAT: IEEE single-precision floating point (4-byte) format in units of seconds. Example: 0x45610000 for 1 hour (1:00:00).
- **User-Defined Events (EV1 to EV8)**
  ASCII: The character strings that indicate that conditions are met and that conditions are not met (the default values are “True” and “False”)
  FLOAT: 0x3F800000 (1) when the conditions are met and 0x00000000 (0) when the conditions are not met
- **No items (NONE)**
  ASCII: NAN (Not A Number)
  FLOAT: 0x7E951BEE (9.91E+37)
- **Data update status**
  ASCII: <NR1> format in units of seconds.
  FLOAT: IEEE single-precision floating point (4-byte) format in units of seconds.
- **Measurement range**
  ASCII: <NR3> format (mantissa: up to 4 digits, exponent: 2 digits. Example: 1.000E+03 for 1000 V range.)
  FLOAT: IEEE single-precision floating point (4-byte) format
- **Other**
  ASCII: <NR3> format (mantissa: up to 5 digits, exponent: 2 digits. Example: [-]123.45.456E+00)
  FLOAT: IEEE single-precision floating point (4-byte) format

Note

- In 180° (Lead/Lag) display, the phase differences Φ (PHI) of elements 1 to 6 are output in the range between -180.00 to 180.00 with lead (D) and lag (G) set to negative and positive values, respectively.
- There may be up to six digits in the mantissa of the Σ of power values P, S, and Q depending on the combination of the voltage range and current range (e.g. the power range). For the table of power ranges, see the features guide, IM WT1801R-01EN.
- The following values always have three decimal places: efficiency values ETA1, ETA2, ETA3, and ETA4; harmonic distortion factor values UHDFk, IHDFk, and PHDFk; and distortion factor values UTHD, ITHD, and PTHD. If the values exceed 100 %, the mantissa will have six digits.

Error data

- **Data does not exist (the display shows "---------")**
  ASCII: NAN (Not A Number)
  FLOAT: 0x7E951BEE (9.91E+37)
- **Over-range (the display shows "---O L---")**
- **Overflow (the display shows "---O F---")**
- **Data over (the display shows " Error ")**
  ASCII: INF (INFinity)
  FLOAT: 0x7E94F56A (9.9E+37)

Preset patterns for numeric data items

This list indicates the measurement function and element that are assigned to each item number (ITEM<x>).

Items that are not set to be measured are displayed or output in the same fashion as when the data does not exist.

Example If the wiring system is not set, the output of ITEM61 in pattern 1 is the same as the output when the data does not exist (NAN if the data format is ASCII).

* “Numeric data functions”(page 5-44) in the DISPlay Group section contains a list of the function names <Function> used in commands (where the command syntax contains) and the function names in the instrument’s menus that correspond to them.

Preset patterns for numeric data items

Applicable commands

```
:NUMeric[:NORMal]:PRESet
```

Pattern 1

ITEM<x>	<Function>	<Element>
1	URMS	1
2	IRMS	1
3	P	1
4	S	1
5	Q	1
6	LAMBda	1
7	PHI	1
8	FU	1
9	FI	1
10	NONE	
11 to 19	URMS to FI	2
20	NONE	
21 to 29	URMS to FI	3
30	NONE	
31 to 39	URMS to FI	4
40	NONE	
41 to 49	URMS to FI	5
50	NONE	
51 to 59	URMS to FI	6
60	NONE	
61 to 69	URMS to FI	SIGMA
70	NONE	
71 to 79	URMS to FI	SIGMB
80	NONE	
81 to 89	URMS to FI	SIGMC
90	NONE	
91 to 1000	NONE	

Pattern2

ITEM<x>	<Function>	<Element>
1	URMS	1
2	UMN	1
3	UDC	1
4	UAC	1
5	IRMS	1
6	IMN	1
7	IDC	1
8	IAC	1
9	P	1
10	S	1
11	Q	1
12	LAMBda	1
13	PHI	1
14	FU	1
15	FI	1
16 to 30	URMS to FI	2
31 to 45	URMS to FI	3
46 to 60	URMS to FI	4
61 to 75	URMS to FI	5
76 to 90	URMS to FI	6
91 to 105	URMS to FI	SIGMA
106 to 120	URMS to FI	SIGMB
121 to 135	URMS to FI	SIGMC
136 to 1000	NONE	

Pattern 3

ITEM<x>	<Function>	<Element>
1	URMS	1
2	UMN	1
3	UDC	1
4	UAC	1
5	IRMS	1
6	IMN	1
7	IDC	1
8	IAC	1
9	P	1
10	S	1
11	Q	1
12	LAMBda	1
13	PHI	1
14	FU	1
15	FI	1
16	UPPeak	1
17	UMPeak	1
18	IPPeak	1
19	IMPeak	1
20	NONE	
21 to 39	URMS to IMPeak	2
40	NONE	
41 to 59	URMS to IMPeak	3
60	NONE	
61 to 79	URMS to IMPeak	4
80	NONE	
81 to 99	URMS to IMPeak	5
100	NONE	
101 to 119	URMS to IMPeak	6
120	NONE	
121 to 139	URMS to IMPeak	SIGMA
140	NONE	
141 to 159	URMS to IMPeak	SIGMB
160	NONE	
161 to 179	URMS to IMPeak	SIGMC
180	NONE	
181 to 1000	NONE	

Pattern 4

ITEM<x>	<Function>	<Element>
1	URMS	1
2	UMN	1
3	UDC	1
4	UAC	1
5	IRMS	1
6	IMN	1
7	IDC	1
8	IAC	1
9	P	1
10	S	1
11	Q	1
12	FU	1
13	FI	1
14	TIME	1
15	WH	1
16	WHP	1
17	WHM	1
18	AH	1
19	AHP	1
20	AHM	1
21 to 40	URMS to AHM	2
41 to 60	URMS to AHM	3
61 to 80	URMS to AHM	4
81 to 100	URMS to AHM	5
101 to 120	URMS to AHM	6
121 to 140	URMS to AHM	SIGMA
141 to 160	URMS to AHM	SIGMB
161 to 180	URMS to AHM	SIGMC
181 to 1000	NONE	

Preset patterns for harmonic measurement numeric list data output items

Applicable commands

```
:NUMeric:LIST:PRESet
```

Pattern 1

ITEM<x>	<Function>	<Element>
1	U	1
2	I	1
3	P	1
4 to 6	U to P	2
7 to 9	U to P	3
10 to 12	U to P	4
13 to 15	U to P	5
16 to 18	U to P	6
19 to 64	NONE	

Pattern 2

ITEM<x>	<Function>	<Element>
1	U	1
2	I	1
3	P	1
4	PHIU	1
5	PHII	1
6 to 10	U to PHII	2
11 to 15	U to PHII	3
16 to 20	U to PHII	4
21 to 25	U to PHII	5
26 to 30	U to PHII	6
31 to 64	NONE	

Pattern 3

ITEM<x>	<Function>	<Element>
1	U	1
2	I	1
3	P	1
4	S	1
5	Q	1
6	LAMBda	1
7	PHI	1
8	PHIU	1
9	PHII	1
10 to 18	U to PHII	2
19 to 27	U to PHII	3
28 to 36	U to PHII	4
37 to 45	U to PHII	5
46 to 54	U to PHII	6
55 to 64	NONE	

Pattern 4

ITEM<x>	<Function>	<Element>
1	U	1
2	I	1
3	P	1
4	Q	1
5	Z	1
6	RS	1
7	XS	1
8	RP	1
9	XP	1
10 to 18	U to XP	2
19 to 27	U to XP	3
28 to 36	U to XP	4
37 to 45	U to XP	5
46 to 54	U to XP	6
55 to 64	NONE	

Preset patterns for high speed data capturing mode numeric data output items

Applicable commands

```
:NUMeric:HSPeed:PRESet
```

ITEM<x>	<Function>	<Element>
1	U	1
2	I	1
3	P	1
4 to 6	U to P	2
7 to 9	U to P	3
10 to 12	U to P	4
13 to 15	U to P	5
16 to 18	U to P	6
19 to 21	U to P	SIGMA
22 to 24	U to P	SIGMB
25 to 27	U to P	SIGMC
28 to 30	NONE	

5.18 RATE Group

The command in this group deals with the data update rate.
You can make the same settings and queries that you can make by pressing UPDATE RATE on the front panel.

:RATE[:RATE]

Sets or queries the data update interval.

Syntax `:RATE[:RATE] {<Time>|AUTO}`
`:RATE[:RATE]?`
<Time> = 50, 100, 200, 500 (ms),
1, 2, 5, 10, 20 (s)

Example `:RATE 500MS`
`:RATE? -> :RATE 500.0E-03`

Description
- The [:RATE] portion in the syntax can be omitted.
- By default, the header response to queries is in the abbreviated form (:COMMunicate:VERBose OFF), so the response will be as shown in the above example.
- If not in abbreviated form (:COMMunicate:VERBose ON), the response will be as shown in the following example.
  `:RATE? -> :RATE:RATE 500.0E-03`
- If the waveform display is enabled and the trigger mode is set to Auto or Normal, the data update interval depends on the trigger operation.

:RATE:AUTO?

Queries all applicable settings for when the data update interval is set to Auto.

Syntax `:RATE:AUTO?`

:RATE:AUTO:TIMeout

Sets or queries the timeout for when the data update interval is set to Auto.

Syntax `:RATE:AUTO:TIMeout {<NRf>}`
`:RATE:AUTO:TIMeout?`
<NRf> = 1, 5, 10, 20 (s)

Example `:RATE:AUTO:TIMEOUT 1`
`:RATE:AUTO:TIMEOUT?`
`-> :RATE:AUTO:TIMEOUT 1`

5.19 STATus Group

The commands in this group are used to make settings and queries related to the status report. There are no front panel keys that correspond to the commands in this group. For information about status reports, see chapter 6.

:STATus?

Queries all the settings for the communication status feature.

Syntax :STATus?

:STATus:CONDition?

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:EESE

Sets or queries the extended event enable register.

Syntax :STATus:EESE <Register>
:STATus:EESE?
<Register> = 0 to 65535

Example :STATUS:EESE #B0000000000000000
:STATus:EESE? -> :STATUS:EESE 0

Description For information about the extended event enable register, see chapter 6, “Status Reports.”

:STATus:EESR?

Queries the contents of the extended event register and clears the register.

Syntax :STATus:EESR?

Example :STATUS:EESR? -> 0

Description For information about the extended event register, see chapter 6, “Status Reports.”

:STATus:ERRor?

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,"Underfined Header"

Description
- If no error has occurred, 0, “Noerror” is returned.
- You can use the :STATus:QMESsage command to specify whether the message is included.

:STATus:FILTer<x>

Sets or queries the transition filter.

Syntax :STATus:FILTer<x> {RISE|FALL|BOTH|NEVer}
:STATus:FILTer<x>?
<x> = 1 to 16

Example :STATUS:FILTER2 RISE
:STATus:FILTER2?
-> :STATUS:FILTER2 RISE

Description
- Set how each bit in the condition register must change to trigger the setting of an event. If a bit is set to RISE, an event is set when the bit changes from 0 to 1.
- For information about the transition filter, see chapter 6, “Status Reports.”

:STATus:QENable

Sets or queries whether messages other than errors will be stored to the error queue (ON/OFF).

Syntax :STATus:QENable {<Boolean>}
:STATus:QENable?

Example :STATUS:QENABLE ON
:STATus:QENABLE? -> :STATus:QENABLE 1

:STATus:QMESsage

Sets or queries whether message information will be attached to the response to the :STATus:ERRor? query (ON/OFF).

Syntax :STATus:QMESsage {<Boolean>}
:STATus:QMESsage?

Example :STATUS:QMESSAGE ON
:STATus:QMESSAGE?
-> :STATus:QMESSAGE 1

:STATus:SPOLl?

Executes serial polling.

Syntax :STATus:SPOLl?

Example :STATUS:SPOLL? -> :STATUS:SPOLL 0

5.20 STORe Group

The commands in this group deal with storage.
You can make the same settings and queries that you can make by pressing STORE START, STORE STOP, STORE RESET (SHIFT+STORE STOP), and STORE SET (SHIFT+STORE START) on the front panel.

:STORe?

Queries all numeric data storage settings.

Syntax :STORe?

:STORe:COUNt

Sets or queries the storage count.

Syntax :STORe:COUNt {<NRf>|INFinite}
:STORe:COUNt?
<NRf> = 1 to 9999999
INFinite = No limit

Example :STORE:COUNT 100
:STORE:COUNT? -> :STORE:COUNT 100

:STORe:FILE?

Queries all settings related to the saving of the data stored in this instrument to files.

Syntax :STORe:FILE?

:STORe:FILE:ANAMing

Sets or queries the auto naming feature for saving stored numeric data to files.

Syntax :STORe:FILE:ANAMing {OFF|NUMBering|DATE}
:STORe:FILE:ANAMing?

Example :STORE:FILE:ANAMING NUMBERING
:STORE:FILE:ANAMING?
-> :STORE:FILE:ANAMING NUMBERING

:STORe:FILE:CDIRectory

Changes the directory that stored numeric data is saved to.

Syntax :STORe:FILE:CDIRectory {<String>}
<String> = Directory name

Example :STORE:FILE:CDIRECTORY "STORE"

Description Specify ".." to move up to the parent directory.

:STORe:FILE:CONVert?

Queries all settings related to the conversion of stored numeric data files into CSV format.

Syntax :STORe:FILE:CONVert?

:STORe:FILE:CONVert:ABORt

Aborts the conversion of a numeric data file to CSV format.

Syntax :STORe:FILE:CONVert:ABORt

Example :STORE:FILE:CONVERT:ABORT

:STORe:FILE:CONVert:AUTO

Sets or queries the on/off status of the automatic conversion of stored numeric data files to CSV format.

Syntax :STORe:FILE:CONVert:AUTO {<Boolean>}
:STORe:FILE:CONVert:AUTO?

Example :STORE:FILE:CONVERT:AUTO ON
:STORE:FILE:CONVERT:AUTO?
-> :STORE:FILE:CONVERT:AUTO 1

:STORe:FILE:CONVert:CDIRectory

Changes the directory that stored numeric data (CSV file) is saved to.

Syntax :STORe:FILE:CONVert:
CDIRectory {<String>}
<String> = Directory name

Example :STORE:FILE:CONVERT:
CDIRECTORY "STORE"

Description Specify ".." to move up to the parent directory.

:STORe:FILE:CONVert:DRIVe

Sets the drive that stored numeric data (CSV file) is saved to.

Syntax :STORe:FILE:CONVert:DRIVe {USER|RAM|USB[,<NRf>]|NETWork}
USER = Internal memory drive
RAM = Internal RAM disk
USB = USB memory device drive,
<NRf> = 0 or 1 (drive number)
NETWork = Network drive

Example :STORE:FILE:CONVERT:DRIVE USER

:STORe:FILE:CONVert:EXECute

Converts the specified stored numeric data file to CSV format.

Syntax :STORe:FILE:CONVert:
EXECute {<String>}
<String> = File name

Example :STORE:FILE:CONVERT:EXECUTE "STORE1"

Description
- Specify the file name without an extension.
- This command is an overlap command.

:STORe:FILE:CONVert:FREE?

Queries the free space (in bytes) on the drive that the stored numeric data (CSV file) is saved to.

Syntax :STORe:FILE:CONVert:FREE?

Example :STORE:FILE:CONVERT:FREE? -> 20912128

:STORe:FILE:CONVert:PATH?

Queries the absolute path of the directory that the stored numeric data (CSV file) is saved to.

Syntax :STORe:FILE:CONVert:PATH?

Example :STORE:FILE:CONVERT:PATH?
-> "USB-0/STORE"

:STORe:FILE:CONVert:SPMode

Sets or queries the on/off state of the save location path designation mode of the stored numeric data (CSV path selection mode)

Syntax :STORe:FILE:CONVert:
SPMode {<Boolean>}
:STORe:FILE:CONVert:SPMode?

Example :STORE:FILE:CONVERT:SPMODE ON
:STORE:FILE:CONVERT:SPMODE?
-> :STORE:FILE:CONVERT:SPMODE 1

:STORe:FILE:DRIVe

Sets the drive that stored numeric data is saved to.

Syntax :STORe:FILE:DRIVe {USER|RAM|
USB[,<NRf>]|NETWork}
USER = Internal memory drive
RAM = Internal RAM disk
USB = USB memory device drive;
<NRf> = 0 or 1 (drive number)
NETWork = Network drive

Example :STORE:FILE:DRIVE USER

:STORe:FILE:FREE?

Queries the free space (in bytes) on the drive that the stored numeric data is saved to.

Syntax :STORe:FILE:FREE?

Example :STORE:FILE:FREE? -> 20912128

:STORe:FILE:NAME

Sets or queries the name of the file that stored numeric data is saved to.

Syntax :STORe:FILE:NAME {<String>}
:STORe:FILE:NAME?
<String> = File name

Example :STORE:FILE:NAME "STORE1"
:STORE:FILE:NAME?
-> :STORE:FILE:NAME "STORE1"

:STORe:FILE:PATH?

Queries the absolute path of the directory that the stored numeric data is saved to.

Syntax :STORe:FILE:PATH?

Example :STORE:FILE:PATH? -> "USB-0/STORE"

:STORe:INTerval

Sets or queries the storage interval.

Syntax :STORe:INTerval {<NRf>,<NRf>,<NRf>}
:STORe:INTerval?
First <NRf> = 0 to 99 (hours)
Second <NRf> = 0 to 59 (minutes)
Third <NRf> = 1 to 59 (seconds)

Example :STORE:INTERVAL 0,0,0
:STORE:INTERVAL?
-> :STORE:INTERVAL 0,0,0

Description This command is valid when the storage mode (:STORe:SMODe) is set to MANual, RTIMe, or INTEGrate.

:STORe:NUMeric?

Queries all numeric data storage item settings.

Syntax :STORe:NUMeric?

:STORe:NUMeric:ITEM

Sets or queries the numeric data storage item selection method.

Syntax :STORe:NUMeric:ITEM {DISPlayed|
SELected}
:STORe:NUMeric:ITEM?
DISPlayed = Automatic selection method in which all the items that are displayed on the screen are selected
SELected = Manual selection method

Example :STORE:NUMERIC:ITEM SELECTED
:STORE:NUMERIC:ITEM?
-> :STORE:NUMERIC:ITEM SELECTED

Description The available options are explained below.
DISPlayed = The numeric items that are displayed on the screen are saved to the file.
SELected = The numeric items that are specified with the commands that start with ":STORe:NUMeric:NORMal: . . ." are saved to the file.

:STORe:NUMeric:NORMal?

Queries all numeric data storage item settings (for the manual selection method).

Syntax :STORe:NUMeric:NORMal?

Description This command is valid when the storage item selection method (:STORe:NUMeric:ITEM) is set to SELected (the manual selection method).

:STORe:NUMeric[:NORMal]:ALL

Collectively sets the on/off status of the output of all element functions when numeric data is stored.

Syntax :STORe:NUMeric[:NORMal]:ALL {<Boolean>}

Example :STORE:NUMERIC:NORMAL:ALL ON

:STORe:NUMeric[:NORMal]:{ELEMent<x>|SIGMA|SIGMB|SIGMC}

Sets or queries the on/off status of the output of the specified element or wiring unit ΣA, ΣB, or ΣC when numeric data is stored.

Syntax :STORe:NUMeric[:NORMal]:{ELEMent<x>|SIGMA|SIGMB|SIGMC} {<Boolean>}
:STORe:NUMeric[:NORMal]:{ELEMent<x>|SIGMA|SIGMB|SIGMC}?
<x> = 1 to 6

Example :STORE:NUMERIC:NORMAL:ELEMENT1 ON
:STORE:NUMERIC:NORMAL:ELEMENT1?
-> :STORE:NUMERIC:NORMAL:ELEMENT1 1

:STORe:NUMeric[:NORMal]:<Function>

Sets or queries the on/off status of the specified function's output when numeric data is stored.

Syntax :STORe:NUMeric[:NORMal]:<Function> {<Boolean>}
:STORe:NUMeric[:NORMal]:<Function>?
<Function> = {URMS|IRMS|P|S|Q|...}

Example :STORE:NUMERIC:NORMAL:URMS ON
:STORE:NUMERIC:NORMAL:URMS?
-> :STORE:NUMERIC:NORMAL:URMS 1

Description For information about the options available for <Function>, see "Numeric data functions" in "Function option list" at the end of the DISPlay group on page 5-44.

:STORe:NUMeric[:NORMal]:PRESet<x>

Presets the output on/off pattern of the element functions to be used when numeric data is stored.

Syntax :STORe:NUMeric[:NORMal]:PRESet<x>
<x> = 1 or 2 (preset number)

Example :STORE:NUMERIC:NORMAL:PRESET1

Description For details on the storage item setting patterns that result when the pattern is reset, see the features guide, IM WT1801R-01EN.

:STORe:RESet

Resets the numeric data storage feature.

Syntax :STORe:RESet

Example :STORE:RESET

:STORe:RTIMe?

Queries the storage start and end times for real-time storage mode.

Syntax :STORe:RTIMe?

:STORe:RTIMe:{STARt|END}

Sets or queries the storage start or end time for real-time storage mode.

Syntax :STORe:RTIMe:{STARt|END} {<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>}
:STORe:RTIMe:{STARt|END}?
{<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>} =
2001,1,1,0,0,0 to 2099,12,31,23,59,59
First <NRf> = 2001 to 2099 (year)
Second <NRf> = 1 to 12 (month)
Third <NRf> = 1 to 31 (day)
Fourth <NRf> = 0 to 23 (hour)
Fifth <NRf> = 0 to 59 (minute)
Sixth <NRf> = 0 to 59 (second)

Example :STORE:RTIME:START 2024,1,1,0,0,0
:STORE:RTIME:START?
-> :STORE:RTIME:START 2024,1,1,0,0,0

Description This command is valid when the storage mode (:STORe:SMODe) is set to RTIMe.

:STORe:SASTart

Sets or queries whether numeric data is stored when storage starts.

Syntax :STORe:SASTart {<Boolean>}
:STORe:SASTart?

Example :STORE:SASTART OFF
:STORE:SASTART? -> :STORE:SASTART 0

Description This command is valid when the storage mode (:STORe:SMODe) is set to MANual, RTIMe, or INTEGrate (and when the storage interval is a value other than 0 for MANual or RTIMe mode).

:STORe:SMODe

Sets or queries the storage mode.

Syntax :STORe:SMODe {MANual|RTIMe|INTEGrate|EVENt|SINGle}
:STORe:SMODe?
MANual = Manual storage mode
RTIMe = Real-time storage mode
INTEGrate = Integration-synchronized storage mode
EVENt = Event-synchronized storage mode
SINGle = Single-shot storage mode

Example :STORE:SMODE MANUAL
:STORE:SMODE? -> :STORE:SMODE MANUAL

:STORe:STARt

Begins the storing of numeric data.

Syntax :STORe:STARt

Example :STORE:START

Description If :STORe:SMODe is set to MANual, the storage operation is executed. If :STORe:SMODe is set to RTIMe, INTEGrate, or EVENt, this instrument enters into a storage wait state. If :STORe:SMODe is set to SINGle, the storage operation is executed.

:STORe:STATe?

Sets or queries the storage state.

Syntax :STORe:STATe?

Example :STORE:STATE? -> RESET

Description The response is as follows:
RESet = Storage reset
READy = Storage standby
STARt = Currently storing
STOP = Storage stopped
COMPlete = Storage complete (or ended due to an error)
CONVert = Converting stored data to CSV format
CLOSe = Final processing of a WTS or HDS file

:STORe:STOP

Stops the storing of numeric data.

Syntax :STORe:STOP

Example :STORE:STOP

:STORe:TEVent

Sets or queries the event that the event-synchronized storage mode will trigger on.

Syntax :STORe:TEVent {<NRf>}
:STORe:TEVent?
<NRf> = 1 to 8 (event number)

Example :STORE:TEVENT 1
:STORE:TEVENT? -> :STORE:TEVENT 1

Description This command is valid when the storage mode (:STORe:SMODe) is set to EVENt.

5.21 SYSTem Group

The commands in this group deal with the system. You can make the same settings and queries that you can make by pressing UTILITY on the front panel and then using the System Config menu.

:SYSTem?

Queries all system settings.

Syntax :SYSTem?

:SYSTem:CLOCk?

Queries all date/time settings.

Syntax :SYSTem:CLOCk?

:SYSTem:CLOCk:DISPlay

Sets or queries the on/off status of the date/time display.

Syntax :SYSTem:CLOCk:DISPlay {<Boolean>}
:SYSTem:CLOCk:DISPlay?

Example :SYSTEM:CLOCK:DISPLAY ON
:SYSTEM:CLOCK:DISPLAY?
-> :SYSTEM:CLOCK:DISPLAY 1

:SYSTem:CLOCk:SNTP?

Queries all settings related to using SNTP to set the date and time.

Syntax :SYSTem:CLOCk:SNTP?

:SYSTem:CLOCk:SNTP[:EXECute]

Uses SNTP to set the date and time.

Syntax :SYSTem:CLOCk:SNTP[:EXECute]

Example :SYSTEM:CLOCK:SNTP:EXECUTE

:SYSTem:CLOCk:SNTP:GMTTime

Sets or queries the time difference from Greenwich Mean Time.

Syntax :SYSTem:CLOCk:SNTP:GMTTime {<String>}
:SYSTem:CLOCk:SNTP:GMTTime?
<String> = "HH:MM" (HH = hours, MM = minutes)

Example :SYSTEM:CLOCK:SNTP:GMTTIME "09:00"
:SYSTEM:CLOCK:SNTP:GMTTIME?
-> :SYSTEM:CLOCK:SNTP:GMTTIME "09:00"

:SYSTem:CLOCk:TYPE

Sets or queries the date/time setup method.

Syntax :SYSTem:CLOCk:TYPE {MANual|SNTP}
:SYSTem:CLOCk:TYPE?

Example :SYSTEM:CLOCK:TYPE MANUAL
:SYSTEM:CLOCK:TYPE?
-> :SYSTEM:CLOCK:TYPE MANUAL

:SYSTem:COMMunicate:COMMand

Sets or queries the communication command type.

Syntax :SYSTem:COMMunicate:COMMand {WT1800R|WT1800E|WT1800|WT1600}
:SYSTem:COMMunicate:COMMand?

Example :SYSTEM:COMMUNICATE:COMMAND WT1800R
:SYSTEM:COMMUNICATE:COMMAND?
-> :SYSTEM:COMMUNICATE:COMMAND WT1800R

Description See the command type settings in chapter 8, “Command Type Compatible with Legacy Models.”

:SYSTem:DATE

Sets or queries the date.

Syntax :SYSTem:DATE {<String>}
:SYSTem:DATE?
<String> = "YY/MM/DD"
(YY = year, MM = month, DD = day)

Example :SYSTEM:DATE "24/01/01"
:SYSTEM:DATE? -> "24/01/01"

Description For year, enter the last two digits of the year according to the Gregorian calendar.

:SYSTem:DFLow:FREQuency

Sets or queries the frequency data display format when a low frequency (or no frequency) input is applied.

Syntax :SYSTem:DFLow:FREQuency {0|ERRor}
:SYSTem:DFLow:FREQuency?

Example :SYSTEM:DFLOW:FREQUENCY ERROR
:SYSTEM:DFLOW:FREQUENCY?
-> :SYSTEM:DFLOW:FREQUENCY ERROR

:SYSTem:DFLow:MOTor

Sets or queries the motor data display format when no pulse is applied.

Syntax :SYSTem:DFLow:MOTor {0|ERRor}
:SYSTem:DFLow:MOTor?

Example :SYSTEM:DFLOW:MOTOR ERROR
:SYSTEM:DFLOW:MOTOR?
-> :SYSTEM:DFLOW:MOTOR ERROR

Description This is only valid on models with the motor evaluation function (/MTR) option.

:SYSTem:DPOint

Sets or queries the type of decimal point that is used when saving various data in ASCII format (CSV).

Syntax :SYSTem:DPOint {PERiod|COMMa}
:SYSTem:DPOint?

Example :SYSTEM:DPOINT PERIOD
:SYSTEM:DPOINT?
-> :SYSTEM:DPOINT PERIOD

:SYSTem:ECLear

Clears error messages displayed on the screen.

Syntax :SYSTem:ECLear

Example :SYSTEM:ECLEAR

:SYSTem:FONT

Sets or queries the menu and message font size.

Syntax :SYSTem:FONT {SMALl|LARGe}
:SYSTem:FONT?

Example :SYSTEM:FONT LARGE
:SYSTEM:FONT?
-> :SYSTEM:FONT LARGE

:SYSTem:KLOCk

Sets or queries the on/off status of the key lock.

Syntax :SYSTem:KLOCk {<Boolean>}
:SYSTem:KLOCk?

Example :SYSTEM:KLOCK OFF
:SYSTEM:KLOCK? -> :SYSTEM:KLOCK 0

:SYSTem:LANGuage?

Queries all display language settings.

Syntax :SYSTem:LANGuage?

:SYSTem:LANGuage:MENU

Sets or queries the menu language.

Syntax :SYSTem:LANGuage:MENU {JAPANese|ENGLish|CHINese|GERMan|RUSSian}
:SYSTem:LANGuage:MENU?

Example :SYSTEM:LANGUAGE:MENU ENGLISH
:SYSTEM:LANGUAGE:MENU?
-> :SYSTEM:LANGUAGE:MENU ENGLISH

:SYSTem:LANGuage:MESSage

Sets or queries the message language.

Syntax :SYSTem:LANGuage:MESSage {JAPANese|ENGLish|CHINese|GERMan|RUSSian}
:SYSTem:LANGuage:MESSage?

Example :SYSTEM:LANGUAGE:MESSAGE ENGLISH
:SYSTEM:LANGUAGE:MESSAGE?
-> :SYSTEM:LANGUAGE:MESSAGE ENGLISH

:SYSTem:LCD?

Queries all LCD settings.

Syntax :SYSTem:LCD?

:SYSTem:LCD:AOFF?

Queries all the settings for the feature that automatically turns off the backlight.

Syntax :SYSTem:LCD:AOFF?

:SYSTem:LCD:AOFF[:STATe]

Sets or queries the on/off status of the feature that automatically turns off the backlight.

Syntax :SYSTem:LCD:AOFF[:STATe] {<Boolean>}
:SYSTem:LCD:AOFF:STATe?

Example :SYSTEM:LCD:AOFF:STATE ON
:SYSTEM:LCD:AOFF:STATE?
-> :SYSTEM:LCD:AOFF:STATE 1

:SYSTem:LCD:AOFF:TIME

Sets or queries the amount of time until the backlight is automatically turned off.

Syntax :SYSTem:LCD:AOFF:TIME {<NRf>}
:SYSTem:LCD:AOFF:TIME?
<NRf> = 1 to 60 (minutes)

Example :SYSTEM:LCD:AOFF:TIME 5
:SYSTEM:LCD:AOFF:TIME?
-> :SYSTEM:LCD:AOFF:TIME 5

:SYSTem:LCD:BRIGhtness

Sets or queries the LCD brightness.

Syntax :SYSTem:LCD:BRIGhtness {<NRf>}
:SYSTem:LCD:BRIGhtness?
<NRf> = 1 to 10

Example :SYSTEM:LCD:BRIGHTNESS 7
:SYSTEM:LCD:BRIGHTNESS?
-> :SYSTEM:LCD:BRIGHTNESS 7

:SYSTem:LCD:COLor?

Queries all LCD color settings.

Syntax :SYSTem:LCD:COLor?

:SYSTem:LCD:COLor:BASecolor

Sets or queries the screen (menu) base color.

Syntax :SYSTem:LCD:COLor:BASecolor {BLUE|GRAY}
:SYSTem:LCD:COLor:BASecolor?

Example :SYSTEM:LCD:COLOR:BASECOLOR BLUE
:SYSTEM:LCD:COLOR:BASECOLOR?
-> :SYSTEM:LCD:COLOR:BASECOLOR BLUE

:SYSTem:LCD:COLor:GRAPh?

Queries all waveform color settings.

Syntax :SYSTem:LCD:COLor:GRAPh?

:SYSTem:LCD:COLor:GRAPh:CHANnel<x>

Sets or queries the specified waveform's color.

Syntax :SYSTem:LCD:COLor:GRAPh:
CHANnel<x> {YELLow|GREen|MAGenta|
CYAN|RED|ORANge|LBLue|PURPle|BLUE|PINK|
LGReen|DBLue|BGReen|SPINk|MGReen|GRAY}
:SYSTem:LCD:COLor:GRAPh:CHANnel<x>?
<x> = 1 to 16 (waveform channel)
YELLow = Yellow
GREen = Green
MAGenta = Magenta
CYAN = Cyan
RED = Red
ORANge = Orange
LBLue = Light blue
PURPle = Purple
BLUE = Blue
PINK = Pink
LGReen = Light green
DBLue = Dark blue
BGReen = Blue green
SPINk = Salmon pink
MGReen = Mild green
GRAY = Gray

Example :SYSTEM:LCD:COLOR:GRAPH:
CHANNEL1 YELLOW
:SYSTEM:LCD:COLOR:GRAPH:CHANNEL1?
-> :SYSTEM:LCD:COLOR:GRAPH:
CHANNEL1 YELLOW

:SYSTem:LCD:COLor:GRAPh:PRESet

Presets the waveform color pattern.

Syntax :SYSTem:LCD:COLor:GRAPh:
PRESet {DEFault|CLASsic}

Example :SYSTEM:LCD:COLOR:GRAPH:
PRESET DEFAULT

:SYSTem:LCD:COLor:INTENsity:GRID

Sets or queries the grid intensity.

Syntax :SYSTem:LCD:COLor:INTENsity:
GRID {<NRf>}
:SYSTem:LCD:COLor:INTENsity:GRID?
<NRf> = 1 to 8 (grid intensity)

Example :SYSTEM:LCD:COLOR:INTENSITY:GRID 4
:SYSTEM:LCD:COLOR:INTENSITY:GRID?
-> :SYSTEM:LCD:COLOR:INTENSITY:GRID 4

:SYSTem:LCD[:STATe]

Sets or queries the on/off status of the backlight.

Syntax :SYSTem:LCD[:STATe] {<Boolean>}
:SYSTem:LCD:STATe?

Example :SYSTEM:LCD:STATE ON
:SYSTEM:LCD:STATE?
-> :SYSTEM:LCD:STATE 1

:SYSTem:MODel?

Queries the model code.

Syntax :SYSTem:MODel?

Example :SYSTEM:MODEL?
-> :SYSTEM:MODEL "WT1806E"

Description Returns the text that appears next to Model on the System Overview screen, which can be accessed by pressing UTILITY.

:SYSTem:RZERo

Sets or queries the on/off status of the rounding to zero feature.

Syntax :SYSTem:RZERo {<Boolean>}
:SYSTem:RZERo?

Example :SYSTEM:RZERO ON
:SYSTEM:RZERO? -> :SYSTEM:RZERO ON

:SYSTem:RESolution

Sets or queries the numeric data display resolution.

Syntax :SYSTem:RESolution {<NRf>}
:SYSTem:RESolution?
<NRf> = 4, 5 (digits)

Example :SYSTEM:RESOLUTION 5
:SYSTEM:RESOLUTION?
-> :SYSTEM:RESOLUTION 5

:SYSTem:SERial?

Queries the serial number.

Syntax :SYSTem:SERial?

Example :SYSTEM:SERIAL?
-> :SYSTEM:SERIAL "123456789"

Description Returns the text that appears next to No. on the System Overview screen, which can be accessed by pressing UTILITY.

:SYSTem:SUFFix?

Queries the suffix code.

Syntax :SYSTem:SUFFix?

Example :SYSTEM:SUFFIX?
-> :SYSTEM:SUFFIX "-5A3-50A3-HE-D/EX6/
B5/G6/V1/DA/MTR/PD2"

Description Returns the text that appears next to Suffix on the System Overview screen, which can be accessed by pressing UTILITY.

:SYSTem:TIME

Sets or queries the time.

Syntax `:SYSTem:TIME {<String>}`
`:SYSTem:TIME?`
<String> = "HH:MM:SS"
(HH = hour, MM = minute, SS = second)

Example `:SYSTEM:TIME "14:30:00"`
`:SYSTEM:TIME? -> "14:30:00"`

:SYSTem:USBKeyboard

Sets or queries the USB keyboard type.

Syntax `:SYSTem:USBKeyboard {JAPANese|ENGLish}`
`:SYSTem:USBKeyboard?`

Example `:SYSTEM:USBKEYBOARD JAPANESE`
`:SYSTEM:USBKEYBOARD?`
`-> :SYSTEM:USBKEYBOARD JAPANESE`

5.22 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?

Queries all waveform display data output settings.

Syntax :WAVeform?

:WAVeform:BYTeorder

Sets or queries the output byte order of the waveform display data (FLOAT format) that is transmitted by the :WAVeform:SEND? command.

Syntax :WAVeform:BYTeorder {LSBFirst|MSBFirst}
:WAVeform:BYTeorder?

Example :WAVEFORM:BYTEORDER MSBFIRST
:WAVEFORM:BYTEORDER?
-> :WAVEFORM:BYTEORDER MSBFIRST

Description This command is valid when :WAVeform:FORMat is set to FLOat.

:WAVeform:END

Sets or queries the output end point of the waveform display data that is transmitted by the :WAVeform:SEND? command.

Syntax :WAVeform:END {<NRf>}
:WAVeform:END?
<NRf> = 0 to 1601

Example :WAVEFORM:END 1601
:WAVEFORM:END?
-> :WAVEFORM:END 1601

:WAVeform:FORMat

Sets or queries the format of the waveform display data that is transmitted by the :WAVeform:SEND? command.

Syntax :WAVeform:FORMat {ASCii|FLOat}
:WAVeform:FORMat?

Example :WAVEFORM:FORMAT FLOAT
:WAVEFORM:FORMAT?
-> :WAVEFORM:FORMAT FLOAT

Description For information about the differences in waveform display data output between formats, see the description of the :WAVeform:SEND? command.

:WAVeform:HOLD

Sets or queries the on/off (hold/release) status of the waveform display data hold feature for all waveforms.

Syntax :WAVeform:HOLD {<Boolean>}
:WAVeform:HOLD?

Example :WAVEFORM:HOLD ON
:WAVEFORM:HOLD? -> :WAVEFORM:HOLD 1

Description
- If :WAVeform:HOLD is set to ON before :WAVeform:SEND? is executed, all the waveform data at that point can be held internally.
- As long as :WAVeform:HOLD is set to ON, waveform data is held even when the waveform display on the screen is updated.
- For example, if you want to acquire U1 and I1 waveform display data at the same point in time, use the following commands:

```
:WAVeform:HOLD ON
:WAVeform:TRACe U1
:WAVeform:SEND?
```
(Receive the waveform display data of U1.)
```
:WAVeform:TRACe I1
:WAVeform:SEND?
```
(Receive the waveform display data of I1.)
```
:WAVeform:HOLD OFF
```

- If :WAVeform:HOLD is set to ON after having already been set to ON before, the waveform display data is cleared, and the most recent waveform data is held internally. When retrieving waveform display data continuously, this method can be used to circumvent the need to repeatedly set :WAVeform:HOLD to OFF.

:WAVeform:LENGth?

Queries the total number of points of the waveform specified by the :WAVeform:TRACe command.

Syntax :WAVeform:LENGth?

Example :WAVEFORM:LENGTH? -> 1602

Description The number of data points is fixed. This command always returns 1602.

:WAVeform:SEND?

Queries the waveform display data specified by the :WAVeform:TRACe command.

Syntax :WAVeform:SEND?

Example
- When :WAVeform:FORMat is set to ASCii:
  :WAVEFORM:SEND? -> <NR3>,<NR3>,...
- When :WAVeform:FORMat is set to FLOat:
  :WAVEFORM:SEND?
  -> #4? (4-digit number of bytes)(data byte sequence)

Description The format of the waveform display data that is output varies depending on how the :WAVeform:FORMat command is set. The different formats are explained below.
- When the format is set to ASCii:
  Physical values are output in <NR3> format. The points are separated by commas.
- When the format is set to FLOat:
  Physical values are output in IEEE single-precision floating point (4-byte) format. The output byte order of the data of each point follows the order that is set by using the :WAVeform:BYTeorder command.

:WAVeform:SRATe?

Queries the sample rate of the acquired waveform.

Syntax :WAVeform:SRATe?

Example :WAVEFORM:SRATE? -> 32.000E+03

:WAVeform:STARt

Sets or queries the output start point of the waveform display data that is transmitted by the :WAVeform:SEND? command.

Syntax :WAVeform:STARt {<NRf>}
:WAVeform:STARt?
<NRf> = 0 to 1601

Example :WAVEFORM:START 0
:WAVEFORM:START?
-> :WAVEFORM:START 0

:WAVeform:TRACe

Sets or queries the target waveform for the :WAVeform:SEND? command.

Syntax :WAVeform:TRACe {U<x>|I<x>|SPEed|TORQue|AUX<x>}
:WAVeform:TRACe?
U<x> and I<x>'s <x> = 1 to 6 (element)
AUX<x>'s <x> = 1 or 2 (AUX input channel)

Example :WAVEFORM:TRACE U1
:WAVEFORM:TRACE?
-> :WAVEFORM:TRACE U1

Description
- SPEed and TORQue can only be selected on models with the motor evaluation function (/MTR) option.
- AUX<x> can only be selected on models with the auxiliary input (/AUX) option.

:WAVeform:TRIGger?

Queries the trigger position of the acquired waveform.

Syntax :WAVeform:TRIGger?

Example :WAVEFORM:TRIGGER? -> 0

Description Because the trigger position is always at the beginning of the waveform display data, 0 is returned.

5.23 Common Command Group

The commands in this group are defined in IEEE 488.2-1992 and are independent from the instrument's individual functions. There are no front panel keys that correspond to the commands in this group.

*CAL?

Executes zero calibration (zero-level compensation, the same operation as pressing CAL—SHIFT+SINGLE) 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

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

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?

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 register, see section 6.3.

*IDN?

Queries the instrument model.

Syntax `*IDN?`

Example `*IDN?`
`-> YOKOGAWA,WT1806R-5A3-50A3,`
`12AB34567,F1.01`

Description
- The information is returned in this form: <Manufacturer>, <Model>, <Serial no.>, <Firmware version>.
- <Model> is returned in the following format: "seven-digit model code—three-digit 5A element structure—four-digit 50A element structure."For details on the model code and element structure, see "Checking the Package Contents" in the getting started guide, IM WT1801R-03EN.

*OPC

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.
- The :COMMunicate:OPSE command is used to specify the overlap command.
- If *OPC is not the last command of the message, its operation is not guaranteed.

*OPC?

Returns ASCII code 1 if the specified overlap command has finished.

Syntax `*OPC?`

Example `*OPC? -> 1`

Description
- For information about how to synchronize a program using *OPC, see page 4-12.
- The :COMMunicate:OPSE command is used to specify the overlap command.
- If *OPC? is not the last command of the message, its operation is not guaranteed.

*OPT?

Queries the installed options.

Syntax `*OPT?`

Example `*OPT? -> EX6,G5,V1,DA,MTR,PD2`

Description
- Returns the presence of the following options: the external current sensor input option (EX1 for the WT1801R, EX2 for the WT1802E, EX3 for the WT1803E, EX4 for the WT1804E, EX5 for the WT1805E, and EX6 for the WT1806E), the harmonic measurement option (G5), the simultaneous dual harmonic measurement option (G6), the RGB output option (V1), the 20-channel D/A output option (DA), the motor evaluation function (MTR), the auxiliary input option (AUX), and the current sensor power (/PD2).
- If none of the options are installed, ASCII code 0 is returned.
- The *OPT? query must be the last query of a program message.
  An error occurs if there is a query after the *OPT query.

*RST

Initializes the settings.

Syntax `*RST`

Example `*RST`

Description
- Also clears *OPC and *OPC? commands that have been sent.
- All settings except communication settings are reset to their factory default values.

*SRE

Sets or queries the service request enable register.

Syntax `*SRE {<NRf>}`
`*SRE?`
<NRf> = 0 to 255

Example `*SRE 239`
`*SRE? -> 175` (because the bit 6, MSS, setting is ignored)

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 section 6.2.

*STB?

Queries the status byte register.

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 section 6.2.

*TRG

Executes single measurement (the same operation as when SINGLE is pressed).

Syntax `*TRG`

Example `*TRG`

Description A multi-line message GET (Group Execute Trigger) will perform the same operation as this command.

*TST?

Performs a self-test and queries the result.

Syntax `*TST?`

Example `*TST? -> 0`

Description
- The self-test consists of tests of each kind of internal memory.
- This command returns 0 if the self-test is successful and 1 if it is not. If the condition of this instrument prevents the self-test from being executed (this happens for example if this instrument is performing integration or storage), this instrument will return an appropriate error code.
- It takes at most 30 seconds for the test to complete. When receiving a response from this instrument, set the timeout to a relatively large value.

*WAI

Holds the execution of the subsequent command until the completion of the specified overlap command.

Syntax `*WAI`

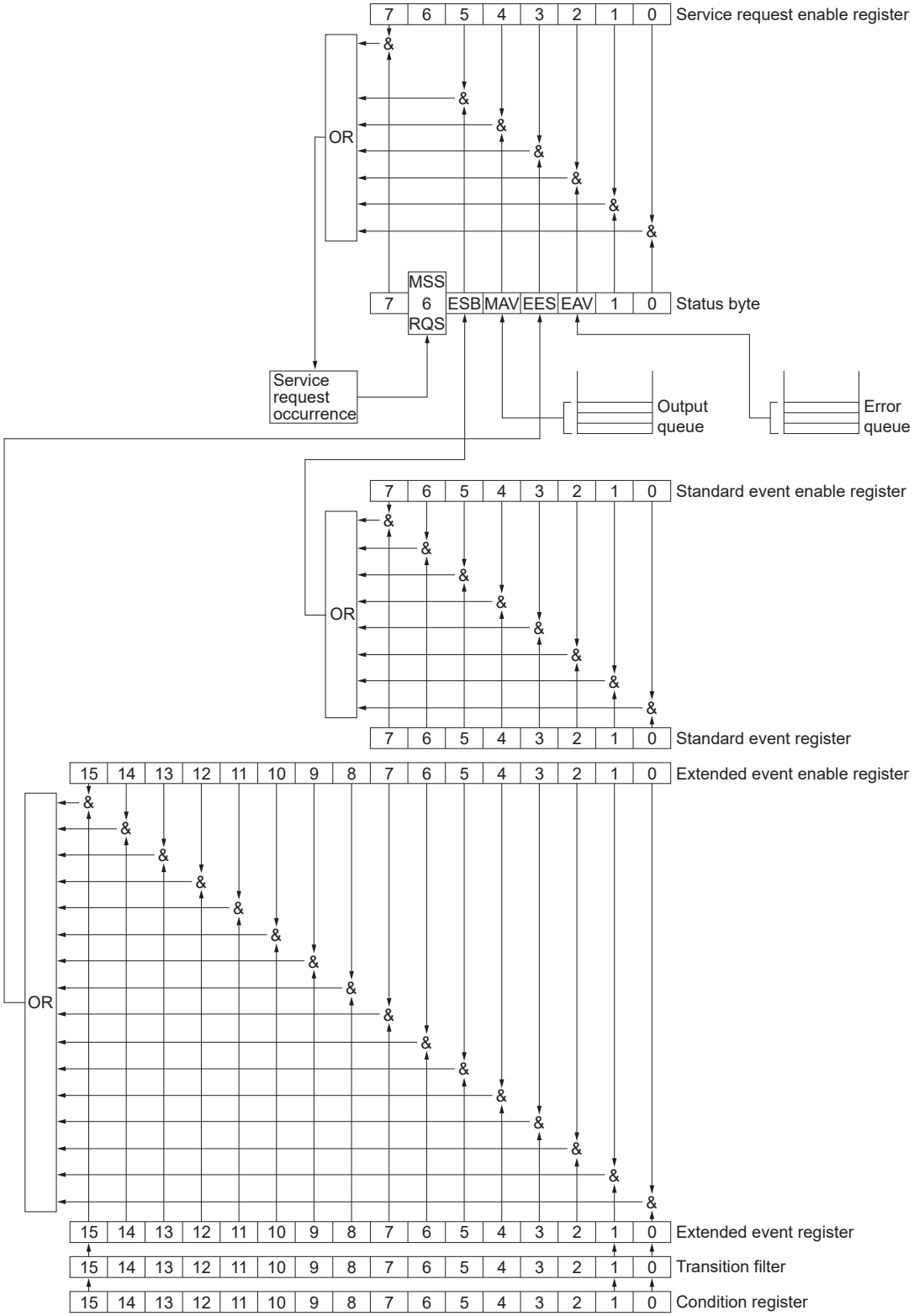
Example `*WAI`

Description
- For information about how to synchronize a program using *WAI, see page 4-11.
- The :COMMunicate:OPSE command is used 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	Writing	Reading
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.

Register	Affected status byte bit
Standard event register	Sets bit 5 (ESB) to 1 or 0
Output queue	Sets bit 4 (MAV) to 1 or 0
Extended event register	Sets bit 3 (EES) to 1 or 0
Error queue	Sets bit 2 (EAV) 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 the bit is set to 1.

Masked register	Mask register
Status byte	Service request enable register
Standard event register	Standard event enable register
Extended event register	Extended event enable 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

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 page 6-6 for details.

Bit 3 EES (Extend Event Summary Bit)

This bit is set to 1 when the logical AND of the extended event register and the extended event enable register is 1. In other words, this bit is set to 1 when a certain event takes place inside the instrument. See page 6-7 for details.

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 page 6-8 for details.

Bit 5 ESB (Event Summary Bit)

This bit is set to 1 when the logical AND of the standard event register and the standard event enable register is 1. In other words, this bit is set to 1 when a certain event takes place inside the instrument. See page 6-5 for details.

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. Use the `*SRE` command for this purpose. 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 this 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 affect 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 set to 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 Command names are misspelled, or character data that is not one of the available options has been received.

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 Parameters are out of range, or a command has been received for an option that is not installed.

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. Use the `*ESE` command for this purpose. 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 is 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 this 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.

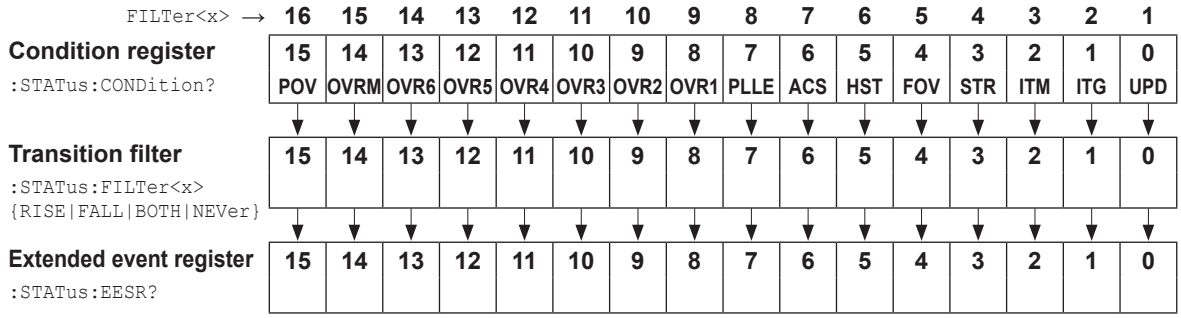
Clearing the standard event register

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	UPD (Updating)	Set to 1 when the measured data is being updated. UPD changing from 1 to 0 indicates that updating has been completed.
Bit 1	ITG (Integrate Busy)	Set to 1 during integration.
Bit 2	ITM (Integrate Timer Busy)	Set to 1 when the integration timer is running.
Bit 3	STR (Store Busy)	Set to 1 during storage.
Bit 4	FOV (Frequency Over)	Set to 1 when a frequency error occurs.
Bit 5	HST (History)	Set to 1 when the history data is being updated. HST changing from 1 to 0 indicates that updating has been completed. * See section "Appendix 2 History Feature".
Bit 6	ACS (Accessing)	This bit is 1 when a drive is being accessed.
Bit 7	PLLE (PLL Source Input Error)	Set to 1 when, during harmonic measurement, there is no PLL source or synchronization cannot be achieved.
Bit 8	OVR1 (Element1 Measured Data Over)	Set to 1 when the voltage or current of element 1 exceeds its range.
Bit 9	OVR2 (Element2 Measured Data Over)	Set to 1 when the voltage or current of element 2 exceeds its range.
Bit 10	OVR3 (Element3 Measured Data Over)	Set to 1 when the voltage or current of element 3 exceeds its range.
Bit 11	OVR4 (Element4 Measured Data Over)	Set to 1 when the voltage or current of element 4 exceeds its range.
Bit 12	OVR5 (Element5 Measured Data Over)	Set to 1 when the voltage or current of element 5 exceeds its range.
Bit 13	OVR6 (Element6 Measured Data Over)	Set to 1 when the voltage or current of element 6 exceeds its range.
Bit 14	OVRM (Motor Measured Data Over)	Set to 1 when the speed or torque of the motor input or auxiliary input AUX1 or AUX2 exceeds its range.
Bit 15	POV (ElementX Input Peak Over)	Set to 1 when a peak over-range is detected on any of the elements.

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 1 when the corresponding condition register bit changes from 0 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

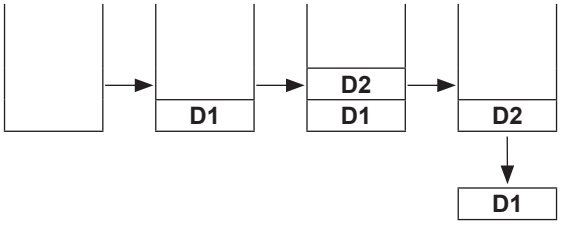
Output queue

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 or not 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 the error queue is empty by checking bit 2 (EAV) in the status byte.

7.1 Overview of Modbus/TCP Communication

Modbus/TCP is one of the communication protocols used to communicate with PCs, PLCs (sequencers), and the like using the TCP/IP protocol over Ethernet networks. This communication protocol is used to read and write to the instrument's internal registers and exchange data with connected devices.
Hereafter, host devices such as PCs will be referred to as client devices.
This instrument can connect to an IEEE802.3 compliant network (1000BASE-T/100BASE-TX).
For communication via the Modbus/TCP protocol, port number 502 is generally used.

Component names and functions

For the names and functions of Ethernet interface components used in Modbus/TCP communication, see section 1.1.

Modbus/TCP function and specifications

Ethernet interface specifications used in Modbus/TCP communication

Item	Specifications
Ports	1
Connector type	RJ-45
Electrical and mechanical	IEEE802.3 compliant
Transmission system	Ethernet 1000BASE-T/100BASE-TX
Communication protocol	TCP/IP
Supported services	DHCP, DNS, SNTP client, FTP server/client, remote control (VXI-11, Modbus/TCP)
Port number	VXI-11: 11024/tcp Modbus/TCP: 502/tcp

This instrument runs as a Modbus server. The number of simultaneous connections is 1.

Connection procedure

Connect an STP (shielded twisted-pair) cable that is connected to a hub or other network device to the Ethernet port on the instrument's rear panel. For details, see section 1.3.

Instrument configuration

TCP/IP configuration is required. For details, see section 1.4.

7.2 Communication with Client Devices

Function codes

This instrument supports the following function codes.

Code	Function	Description
03	Reads the hold register	Continuous reading from 0001 to 0010 is possible.
04	Reads the input register	Up to 125 values can be read continuously from 0001 to 5972.
06	Writes to the hold register	Writing is possible only to one register in the range of 0001 to 0010.

Specifying registers

Registers are specified from a client device in the following manner.

- If an off-the-shelf SCADA or the like is used, specify the reference number (Ref No.) listed in section 7.3 , “Register Functions and Applications.”
- For communication programs that you create, specify the relative number (H No.) listed in section 7.3, “Register Functions and Applications.”

Example: To specify the rms voltage data of element 1 (float lower bytes) (input register: 0101)

- For a request using an off-the-shelf SCADA or the like, specify Ref No. 30101.
- For a request using a communication program that you create, specify H No. 0064.

7.3 Register Functions and Applications

Measured data, setup data, and other types of data of this instrument are assigned to the internal registers for Modbus/TCP. A client device can send commands to this instrument using Modbus/TCP communication to read and write to the internal registers of this instrument. This enables measured data and the like to be retrieved and the instrument to be controlled such as starting integration.

Register configuration

	Register number	Group	Description
Input register	0001 to 0070	Measured data, status	Data not dependent on element/Σ, status
	0071 to 0094	Measurement range information	Voltage and current ranges of each element
	0101 to 0200	Measured data (element 1)	Normal and harmonic measurement data of element 1
	0201 to 0300	Measured data (element 2)	Normal and harmonic measurement data of element 2
	0301 to 0400	Measured data (element 3)	Normal and harmonic measurement data of element 3
	0400 to 0500	Measured data (element 4)	Normal and harmonic measurement data of element 4
	0501 to 0600	Measured data (element 5)	Normal and harmonic measurement data of element 5
	0601 to 0700	Measured data (element 6)	Normal and harmonic measurement data of element 6
	1001 to 1100	Measured data (ΣA)	Normal measurement data of wiring unit ΣA
	1101 to 1200	Measured data (ΣB)	Normal measurement data of wiring unit ΣB
	1201 to 1300	Measured data (ΣC)	Normal measurement data of wiring unit ΣC
	1501 to 1538	Measured data (motor)	Measurement data of motor input
	1601 to 1608	Measured data (AUX)	Measurement data of AUX input
	2001 to 4000	Communication output item data	Measured data synchronized with communication output item settings (**:NUMeric[:NORMal]:ITEM<x>** command)
	4001 to 4096	Numeric display item data (4-value display)	Measured data synchronized with display item settings Numeric display (Numeric): 4-value display
	4101 to 4292	Numeric display item data (8-value display)	Measured data synchronized with display item settings Numeric display (Numeric): 8-value display
	4301 to 4684	Numeric display item data (16-value display)	Measured data synchronized with display item settings Numeric display (Numeric): 6-value display
	5001 to 5972	Numeric display item data (matrix display)	Measured data synchronized with display item settings Numeric display (Numeric): matrix display
Hold register	0001 to 0010	Control data	Hold register values, control integration operation
Other		Reserved area (blank)	Not to be used. Operation not guaranteed if written to.

Register map (input register)

Reg No.	Ref No.	H No.	Register name		Register description		Notes
Data not dependent on status, element, or wiring unit							
0001	30001	0000	Update Count		Data update counter	(uint 16)	0 to 65535
0002	30002	0001	Peak Over		Peak over-range status	(uint 16)	0 to 65535
0003	30003	0002	Eta1	L	Efficiency 1	(float, lower 2 bytes)	
0004	30004	0003		H		(float, upper 2 bytes)	
0005	30005	0004	Eta2	L	Efficiency 2	(float, lower 2 bytes)	
0006	30006	0005		H		(float, upper 2 bytes)	
0007	30007	0006	Eta3	L	Efficiency 3	(float, lower 2 bytes)	
0008	30008	0007		H		(float, upper 2 bytes)	
0009	30009	0008	Eta4	L	Efficiency 4	(float, lower 2 bytes)	
0010	30010	0009		H		(float, upper 2 bytes)	
0011	30011	000A	F1	L	User-defined function 1	(float, lower 2 bytes)	
0012	30012	000B		H		(float, upper 2 bytes)	
0013	30013	000C	F2	L	User-defined function 2	(float, lower 2 bytes)	
0014	30014	000D		H		(float, upper 2 bytes)	
0015	30015	000E	F3	L	User-defined function 3	(float, lower 2 bytes)	
0016	30016	000F		H		(float, upper 2 bytes)	
0017	30017	0010	F4	L	User-defined function 4	(float, lower 2 bytes)	
0018	30018	0011		H		(float, upper 2 bytes)	
0019	30019	0012	F5	L	User-defined function 5	(float, lower 2 bytes)	
0020	30020	0013		H		(float, upper 2 bytes)	
0021	30021	0014	F6	L	User-defined function 6	(float, lower 2 bytes)	
0022	30022	0015		H		(float, upper 2 bytes)	
0023	30023	0016	F7	L	User-defined function 7	(float, lower 2 bytes)	
0024	30024	0017		H		(float, upper 2 bytes)	
0025	30025	0018	F8	L	User-defined function 8	(float, lower 2 bytes)	
0026	30026	0019		H		(float, upper 2 bytes)	
0027	30027	001A	F9	L	User-defined function 9	(float, lower 2 bytes)	
0028	30028	001B		H		(float, upper 2 bytes)	
0029	30029	001C	F10	L	User-defined function 10	(float, lower 2 bytes)	
0030	30030	001D		H		(float, upper 2 bytes)	
0031	30031	001E	F11	L	User-defined function 11	(float, lower 2 bytes)	
0032	30032	001F		H		(float, upper 2 bytes)	
0033	30033	0020	F12	L	User-defined function 12	(float, lower 2 bytes)	
0034	30034	0021		H		(float, upper 2 bytes)	
0035	30035	0022	F13	L	User-defined function 13	(float, lower 2 bytes)	
0036	30036	0023		H		(float, upper 2 bytes)	
0037	30037	0024	F14	L	User-defined function 14	(float, lower 2 bytes)	
0038	30038	0025		H		(float, upper 2 bytes)	
0039	30039	0026	F15	L	User-defined function 15	(float, lower 2 bytes)	
0040	30040	0027		H		(float, upper 2 bytes)	
0041	30041	0028	F16	L	User-defined function 16	(float, lower 2 bytes)	
0042	30042	0029		H		(float, upper 2 bytes)	
0043	30043	002A	F17	L	User-defined function 17	(float, lower 2 bytes)	
0044	30044	002B		H		(float, upper 2 bytes)	
0045	30045	002C	F18	L	User-defined function 18	(float, lower 2 bytes)	
0046	30046	002D		H		(float, upper 2 bytes)	
0047	30047	002E	F19	L	User-defined function 19	(float, lower 2 bytes)	
0048	30048	002F		H		(float, upper 2 bytes)	
0049	30049	0030	F20	L	User-defined function 20	(float, lower 2 bytes)	
0050	30050	0031		H		(float, upper 2 bytes)	

Reg No.	Ref No.	H No.	Register name		Register description		Notes
0051	30051	0032	Ev1	L	User-defined event 1	(float, lower 2 bytes)	
0052	30052	0033		H		(float, upper 2 bytes)	
0053	30053	0034	Ev2	L	User-defined event 2	(float, lower 2 bytes)	
0054	30054	0035		H		(float, upper 2 bytes)	
0055	30055	0036	Ev3	L	User-defined event 3	(float, lower 2 bytes)	
0056	30056	0037		H		(float, upper 2 bytes)	
0057	30057	0038	Ev4	L	User-defined event 4	(float, lower 2 bytes)	
0058	30058	0039		H		(float, upper 2 bytes)	
0059	30059	003A	Ev5	L	User-defined event 5	(float, lower 2 bytes)	
0060	30060	003B		H		(float, upper 2 bytes)	
0061	30061	003C	Ev6	L	User-defined event 6	(float, lower 2 bytes)	
0062	30062	003D		H		(float, upper 2 bytes)	
0063	30063	003E	Ev7	L	User-defined event 7	(float, lower 2 bytes)	
0064	30064	003F		H		(float, upper 2 bytes)	
0065	30065	0040	Ev8	L	User-defined event 8	(float, lower 2 bytes)	
0066	30066	0041		H		(float, upper 2 bytes)	
0067	30067	0042	fPLL1	L	Frequency of PLL source 1	(float, lower 2 bytes)	
0068	30068	0043		H		(float, upper 2 bytes)	
0069	30069	0044	fPLL2	L	Frequency of PLL source 2	(float, lower 2 bytes)	
0070	30070	0045		H		(float, upper 2 bytes)	

Measurement range information

0071	30071	0046	URange1	L	Voltage 1 range value	(float, lower 2 bytes)	
0072	30072	0047		H		(float, upper 2 bytes)	
0073	30073	0048	IRange1	L	Current 1 range value	(float, lower 2 bytes)	
0074	30074	0049		H		(float, upper 2 bytes)	
0075	30075	004A	URange2	L	Voltage 2 range value	(float, lower 2 bytes)	
0076	30076	004B		H		(float, upper 2 bytes)	
0077	30077	004C	IRange2	L	Current 2 range value	(float, lower 2 bytes)	
0078	30078	004D		H		(float, upper 2 bytes)	
0079	30079	004E	URange3	L	Voltage 3 range value	(float, lower 2 bytes)	
0080	30080	004F		H		(float, upper 2 bytes)	
0081	30081	0050	IRange3	L	Current 3 range value	(float, lower 2 bytes)	
0082	30082	0051		H		(float, upper 2 bytes)	
0083	30083	0052	URange4	L	Voltage 4 range value	(float, lower 2 bytes)	
0084	30084	0053		H		(float, upper 2 bytes)	
0085	30085	0054	IRange4	L	Current 4 range value	(float, lower 2 bytes)	
0086	30086	0055		H		(float, upper 2 bytes)	
0087	30087	0056	URange5	L	Voltage 5 range value	(float, lower 2 bytes)	
0088	30088	0057		H		(float, upper 2 bytes)	
0089	30089	0058	IRange5	L	Current 5 range value	(float, lower 2 bytes)	
0090	30090	0059		H		(float, upper 2 bytes)	
0091	30091	005A	URange6	L	Voltage 6 range value	(float, lower 2 bytes)	
0092	30092	005B		H		(float, upper 2 bytes)	
0093	30093	005C	IRange6	L	Current 6 range value	(float, lower 2 bytes)	
0094	30094	005D		H		(float, upper 2 bytes)	

Element 1 normal measurement data

0101	30101	0064	Urms1	L	True rms voltage 1	(float, lower 2 bytes)	
0102	30102	0065		H		(float, upper 2 bytes)	
0103	30103	0066	Irms1	L	True rms current 1	(float, lower 2 bytes)	
0104	30104	0067		H		(float, upper 2 bytes)	
0105	30105	0068	P1	L	Active power 1	(float, lower 2 bytes)	
0106	30106	0069		H		(float, upper 2 bytes)	
0107	30107	006A	S1	L	Apparent power 1	(float, lower 2 bytes)	
0108	30108	006B		H		(float, upper 2 bytes)	

Reg No.	Ref No.	H No.	Register name		Register description		Notes
0109	30109	006C	Q1	L	Reactive power 1	(float, lower 2 bytes)	
0110	30110	006D		H		(float, upper 2 bytes)	
0111	30111	006E	Lambda1	L	Power factor (λ) 1	(float, lower 2 bytes)	
0112	30112	006F		H		(float, upper 2 bytes)	
0113	30113	0070	Phi1	L	Phase difference (φ) 1	(float, lower 2 bytes)	
0114	30114	0071		H		(float, upper 2 bytes)	
0115	30115	0072	fU1	L	Voltage 1 frequency	(float, lower 2 bytes)	
0116	30116	0073		H		(float, upper 2 bytes)	
0117	30117	0074	fI1	L	Current 1 frequency	(float, lower 2 bytes)	
0118	30118	0075		H		(float, upper 2 bytes)	
0119	30119	0076	U+pk1	L	Maximum voltage 1	(float, lower 2 bytes)	
0120	30120	0077		H		(float, upper 2 bytes)	
0121	30121	0078	U-pk1	L	Minimum voltage 1	(float, lower 2 bytes)	
0122	30122	0079		H		(float, upper 2 bytes)	
0123	30123	007A	I+pk1	L	Maximum current 1	(float, lower 2 bytes)	
0124	30124	007B		H		(float, upper 2 bytes)	
0125	30125	007C	I-pk1	L	Minimum current 1	(float, lower 2 bytes)	
0126	30126	007D		H		(float, upper 2 bytes)	
0127	30127	007E	P+pk1	L	Maximum power 1	(float, lower 2 bytes)	
0128	30128	007F		H		(float, upper 2 bytes)	
0129	30129	0080	P-pk1	L	Minimum power 1	(float, lower 2 bytes)	
0130	30130	0081		H		(float, upper 2 bytes)	
0131	30131	0082	Time1	L	Integration time 1	(float, lower 2 bytes)	
0132	30132	0083		H		(float, upper 2 bytes)	
0133	30133	0084	WP1	L	Sum of positive and negative watt hours 1	(float, lower 2 bytes)	
0134	30134	0085		H		(float, upper 2 bytes)	
0135	30135	0086	WP+1	L	Positive watt hours 1	(float, lower 2 bytes)	
0136	30136	0087		H		(float, upper 2 bytes)	
0137	30137	0088	WP-1	L	Negative watt hours 1	(float, lower 2 bytes)	
0138	30138	0089		H		(float, upper 2 bytes)	
0139	30139	008A	q1	L	Sum of positive and negative ampere hours 1	(float, lower 2 bytes)	
0140	30140	008B		H		(float, upper 2 bytes)	
0141	30141	008C	q+1	L	Positive ampere hour 1	(float, lower 2 bytes)	
0142	30142	008D		H		(float, upper 2 bytes)	
0143	30143	008E	q-1	L	Negative ampere hour 1	(float, lower 2 bytes)	
0144	30144	008F		H		(float, upper 2 bytes)	
0145	30145	0090	WS1	L	Volt-ampere hour 1	(float, lower 2 bytes)	
0146	30146	0091		H		(float, upper 2 bytes)	
0147	30147	0092	WQ1	L	Var hour 1	(float, lower 2 bytes)	
0148	30148	0093		H		(float, upper 2 bytes)	
0149	30149	0094	Umn1	L	Rectified mean voltage calibrated to the rms value 1	(float, lower 2 bytes)	
0150	30150	0095		H		(float, upper 2 bytes)	
0151	30151	0096	Udc1	L	DC voltage 1 (simple average)	(float, lower 2 bytes)	
0152	30152	0097		H		(float, upper 2 bytes)	
0153	30153	0098	Urmn1	L	Rectified mean voltage 1	(float, lower 2 bytes)	
0154	30154	0099		H		(float, upper 2 bytes)	
0155	30155	009A	Uac1	L	AC voltage component 1	(float, lower 2 bytes)	
0156	30156	009B		H		(float, upper 2 bytes)	
0157	30157	009C	Imn1	L	Rectified mean current calibrated to the rms value 1	(float, lower 2 bytes)	
0158	30158	009D		H		(float, upper 2 bytes)	
0159	30159	009E	Idc1	L	DC current 1 (simple average)	(float, lower 2 bytes)	
0160	30160	009F		H		(float, upper 2 bytes)	
0161	30161	00A0	Irmn1	L	Rectified mean current 1	(float, lower 2 bytes)	
0162	30162	00A1		H		(float, upper 2 bytes)	

Reg No.	Ref No.	H No.	Register name		Register description		Notes
0163	30163	00A2	Iac1	L	AC current component 1	(float, lower 2 bytes)	
0164	30164	00A3		H		(float, upper 2 bytes)	
0165	30165	00A4	(CfU1)	L	Voltage 1 crest factor	(float, lower 2 bytes)	
0166	30166	00A5		H		(float, upper 2 bytes)	
0167	30167	00A6	(CfI1)	L	Current 1 crest factor	(float, lower 2 bytes)	
0168	30168	00A7		H		(float, upper 2 bytes)	
0169	30169	00A8	Pc1	L	Corrected power 1	(float, lower 2 bytes)	
0170	30170	00A9		H		(float, upper 2 bytes)	

Element 1 Harmonic measurement data

0171	30171	00AA	U1(Total)	L	Total value of all harmonic components of voltage 1	(float, lower 2 bytes)	
0172	30172	00AB		H		(float, upper 2 bytes)	
0173	30173	00AC	U1(1)	L	1st harmonic value of voltage 1 (fundamental wave)	(float, lower 2 bytes)	
0174	30174	00AD		H		(float, upper 2 bytes)	
0175	30175	00AE	I1(Total)	L	Total value of all harmonic components of current 1	(float, lower 2 bytes)	
0176	30176	00AF		H		(float, upper 2 bytes)	
0177	30177	00B0	I1(1)	L	1st harmonic value of current 1 (fundamental wave)	(float, lower 2 bytes)	
0178	30178	00B1		H		(float, upper 2 bytes)	
0179	30179	00B2	P1(Total)	L	Total value of all harmonic components of active power 1	(float, lower 2 bytes)	
0180	30180	00B3		H		(float, upper 2 bytes)	
0181	30181	00B4	P1(1)	L	1st harmonic value of active power 1 (fundamental wave)	(float, lower 2 bytes)	
0182	30182	00B5		H		(float, upper 2 bytes)	
0183	30183	00B6	S1(Total)	L	Total value of all harmonic components of apparent power 1	(float, lower 2 bytes)	
0184	30184	00B7		H		(float, upper 2 bytes)	
0185	30185	00B8	S1(1)	L	1st harmonic value of apparent power 1 (fundamental wave)	(float, lower 2 bytes)	
0186	30186	00B9		H		(float, upper 2 bytes)	
0187	30187	00BA	Q1(Total)	L	Total value of all harmonic components of reactive power 1	(float, lower 2 bytes)	
0188	30188	00BB		H		(float, upper 2 bytes)	
0189	30189	00BC	Q1(1)	L	1st harmonic value of reactive power 1 (fundamental wave)	(float, lower 2 bytes)	
0190	30190	00BD		H		(float, upper 2 bytes)	
0191	30191	00BE	Lambda1(1)	L	Power factor λ1(1) of the 1st order (fundamental wave)	(float, lower 2 bytes)	
0192	30192	00BF		H		(float, upper 2 bytes)	
0193	30193	00C0	Phi1(1)	L	Phase difference between the voltage and current of the 1st order (fundamental wave), φ1(1)	(float, lower 2 bytes)	
0194	30194	00C1		H		(float, upper 2 bytes)	
0195	30195	00C2	Uthd1	L	Total harmonic distortion of voltage 1	(float, lower 2 bytes)	
0196	30196	00C3		H		(float, upper 2 bytes)	
0197	30197	00C4	Ithd1	L	Total harmonic distortion of current 1	(float, lower 2 bytes)	
0198	30198	00C5		H		(float, upper 2 bytes)	
0199	30199	00C6	Pthd1	L	Total harmonic distortion of active power 1	(float, lower 2 bytes)	
0200	30200	00C7		H		(float, upper 2 bytes)	

Element 2 normal measurement data

0201	30201	00C8	Urms2	L	True rms voltage 2	(float, lower 2 bytes)	
0202	30202	00C9		H		(float, upper 2 bytes)	
to 0270							

Element 2 Harmonic measurement data

0271	30271	010E	U2(Total)	L	Total value of all harmonic components of voltage 2	(float, lower 2 bytes)	
0272	30272	010F		H		(float, upper 2 bytes)	
to 0300							

Element 3 normal measurement data

0301	30301	012C	Urms3	L	True rms voltage 3	(float, lower 2 bytes)	
0302	30302	012D		H		(float, upper 2 bytes)	
to 0370							

Reg No.	Ref No.	H No.	Register name		Register description		Notes
Element 3 Harmonic measurement data							
0371	30371	0172	U3(Total)	L	Total value of all harmonic components of voltage 3	(float, lower 2 bytes)	
0372	30372	0173		H		(float, upper 2 bytes)	
to 0400							
Element 4 normal measurement data							
0401	30401	0190	Urms4	L	True rms voltage 4	(float, lower 2 bytes)	
0402	30402	0191		H		(float, upper 2 bytes)	
to 0470							
Element 4 Harmonic measurement data							
0471	30471	01D6	U4(Total)	L	Total value of all harmonic components of voltage 4	(float, lower 2 bytes)	
0472	30472	01D7		H		(float, upper 2 bytes)	
to 0500							
Element 5 normal measurement data							
0501	30501	01F4	Urms5	L	True rms voltage 5	(float, lower 2 bytes)	
0502	30502	01F5		H		(float, upper 2 bytes)	
to 0570							
Element 5 Harmonic measurement data							
0571	30571	023A	U5(Total)	L	Total value of all harmonic components of voltage 5	(float, lower 2 bytes)	
0572	30572	023B		H		(float, upper 2 bytes)	
to 0600							
Element 6 normal measurement data							
0601	30601	0258	Urms6	L	True rms voltage 6	(float, lower 2 bytes)	
0602	30602	0259		H		(float, upper 2 bytes)	
to 0670							
Element 6 Harmonic measurement data							
0671	30671	029E	U6(Total)	L	Total value of all harmonic components of voltage 6	(float, lower 2 bytes)	
0672	30672	029F		H		(float, upper 2 bytes)	
to 0700							
Wiring unit ΣA normal measurement data and delta computation data							
1001	31001	03E8	UrmsΣA	L	True rms voltage of ΣA	(float, lower 2 bytes)	
1002	31002	03E9		H		(float, upper 2 bytes)	
1003	31003	03EA	IrmsΣA	L	True rms current of ΣA	(float, lower 2 bytes)	
1004	31004	03EB		H		(float, upper 2 bytes)	
1005	31005	03EC	PΣA	L	Active power of ΣA	(float, lower 2 bytes)	
1006	31006	03ED		H		(float, upper 2 bytes)	
1007	31007	03EE	SΣA	L	Apparent power of ΣA	(float, lower 2 bytes)	
1008	31008	03EF		H		(float, upper 2 bytes)	
1009	31009	03F0	QΣA	L	Reactive power of ΣA	(float, lower 2 bytes)	
1010	31010	03F1		H		(float, upper 2 bytes)	
1011	31011	03F2	LambdaΣA	L	Power factor of ΣA (λ)	(float, lower 2 bytes)	
1012	31012	03F3		H		(float, upper 2 bytes)	
1013	31013	03F4	PhiΣA	L	Phase difference of ΣA (φ)	(float, lower 2 bytes)	
1014	31014	03F5		H		(float, upper 2 bytes)	
1015	31015	03F6	DeltaU1	L	ΣA delta voltage computation 1 (ΔU1)	(float, lower 2 bytes)	
1016	31016	03F7		H		(float, upper 2 bytes)	
1017	31017	03F8	DeltaU2	L	ΣA delta voltage computation 2 (ΔU2)	(float, lower 2 bytes)	
1018	31018	03F9		H		(float, upper 2 bytes)	
1019	31019	03FA	DeltaU3	L	ΣA delta voltage computation 3 (ΔU3)	(float, lower 2 bytes)	
1020	31020	03FB		H		(float, upper 2 bytes)	
1021	31021	03FC	DeltaUΣ	L	ΣA delta voltage computation Σ (ΔUΣ)	(float, lower 2 bytes)	
1022	31022	03FD		H		(float, upper 2 bytes)	

Reg No.	Ref No.	H No.	Register name		Register description		Notes
1023	31023	03FE	DeltaI	L	ΣA delta current computation (ΔI)	(float, lower 2 bytes)	
1024	31024	03FF		H		(float, upper 2 bytes)	
1025	31025	0400	DeltaP1	L	ΣA delta power computation 1 (ΔP1)	(float, lower 2 bytes)	
1026	31026	0401		H		(float, upper 2 bytes)	
1027	31027	0402	DeltaP2	L	ΣA delta power computation 2 (ΔP2)	(float, lower 2 bytes)	
1028	31028	0403		H		(float, upper 2 bytes)	
1029	31029	0404	DeltaP3	L	ΣA delta power computation 3 (ΔP3)	(float, lower 2 bytes)	
1030	31030	0405		H		(float, upper 2 bytes)	
1031	31031	0406	DeltaPΣ	L	ΣA delta power computation Σ (ΔPΣ)	(float, lower 2 bytes)	
1032	31032	0407		H		(float, upper 2 bytes)	
1033	31033	0408	WPΣA	L	Sum of positive and negative watt hours of ΣA	(float, lower 2 bytes)	
1034	31034	0409		H		(float, upper 2 bytes)	
1035	31035	040A	WP+ΣA	L	Positive watt hours of ΣA	(float, lower 2 bytes)	
1036	31036	040B		H		(float, upper 2 bytes)	
1037	31037	040C	WP-ΣA	L	Negative watt hours of ΣA	(float, lower 2 bytes)	
1038	31038	040D		H		(float, upper 2 bytes)	
1039	31039	040E	qΣA	L	Sum of positive and negative ampere hours of ΣA	(float, lower 2 bytes)	
1040	31040	040F		H		(float, upper 2 bytes)	
1041	31041	0410	q+ΣA	L	Positive ampere hour of ΣA	(float, lower 2 bytes)	
1042	31042	0411		H		(float, upper 2 bytes)	
1043	31043	0412	q-ΣA	L	Negative ampere hour of ΣA	(float, lower 2 bytes)	
1044	31044	0413		H		(float, upper 2 bytes)	
1045	31045	0414	WSΣA	L	Volt-ampere hours of ΣA	(float, lower 2 bytes)	
1046	31046	0415		H		(float, upper 2 bytes)	
1047	31047	0416	WQΣA	L	Var hours of ΣA	(float, lower 2 bytes)	
1048	31048	0417		H		(float, upper 2 bytes)	
1049	31049	0418	UmnΣA	L	Rectified mean voltage calibrated to the rms value of ΣA	(float, lower 2 bytes)	
1050	31050	0419		H		(float, upper 2 bytes)	
1051	31051	041A	UdcΣA	L	DC voltage of ΣA (Simple average)	(float, lower 2 bytes)	
1052	31052	041B		H		(float, upper 2 bytes)	
1053	31053	041C	UrmnΣA	L	Rectified mean voltage of ΣA	(float, lower 2 bytes)	
1054	31054	041D		H		(float, upper 2 bytes)	
1055	31055	041E	UacΣA	L	AC voltage component of ΣA	(float, lower 2 bytes)	
1056	31056	041F		H		(float, upper 2 bytes)	
1057	31057	0420	ImnΣA	L	Rectified mean current calibrated to the rms value of ΣA	(float, lower 2 bytes)	
1058	31058	0421		H		(float, upper 2 bytes)	
1059	31059	0422	IdcΣA	L	DC current of ΣA (simple average)	(float, lower 2 bytes)	
1060	31060	0423		H		(float, upper 2 bytes)	
1061	31061	0424	IrmnΣA	L	Rectified mean current of ΣA	(float, lower 2 bytes)	
1062	31062	0425		H		(float, upper 2 bytes)	
1063	31063	0426	IacΣA	L	AC current component of ΣA	(float, lower 2 bytes)	
1064	31064	0427		H		(float, upper 2 bytes)	
1065	31065	0428		L		(float, lower 2 bytes)	
1066	31066	0429		H		(float, upper 2 bytes)	
1067	31067	042A		L		(float, lower 2 bytes)	
1068	31068	042B		H		(float, upper 2 bytes)	
1069	31069	042C	PcΣA	L	Corrected power of ΣA	(float, lower 2 bytes)	
1070	31070	042D		H		(float, upper 2 bytes)	

Wiring unit ΣA harmonic measurement data

1071	31071	042E	UΣA(Total)	L	Total value of all harmonic components of voltage ΣA	(float, lower 2 bytes)	
1072	31072	042F		H		(float, upper 2 bytes)	
1073	31073	0430	UΣA(1)	L	1st order harmonic value of voltage ΣA (fundamental wave)	(float, lower 2 bytes)	
1074	31074	0431		H		(float, upper 2 bytes)	

Reg No.	Ref No.	H No.	Register name		Register description		Notes
1075	31075	0432	IΣA(Total)	L	Total value of all harmonic components of current ΣA	(float, lower 2 bytes)	
1076	31076	0433		H		(float, upper 2 bytes)	
1077	31077	0434	IΣA(1)	L	1st harmonic value of current ΣA (fundamental wave)	(float, lower 2 bytes)	
1078	31078	0435		H		(float, upper 2 bytes)	
1079	31079	0436	PΣA(Total)	L	Total value of all harmonic components of active power ΣA	(float, lower 2 bytes)	
1080	31080	0437		H		(float, upper 2 bytes)	
1081	31081	0438	PΣA(1)	L	1st harmonic value of active power ΣA (fundamental wave)	(float, lower 2 bytes)	
1082	31082	0439		H		(float, upper 2 bytes)	
1083	31083	043A	SΣA(Total)	L	Total value of all harmonic components of apparent power ΣA	(float, lower 2 bytes)	
1084	31084	043B		H		(float, upper 2 bytes)	
1085	31085	043C	SΣA(1)	L	1st harmonic value of apparent power ΣA (fundamental wave)	(float, lower 2 bytes)	
1086	31086	043D		H		(float, upper 2 bytes)	
1087	31087	043E	QΣA(Total)	L	Total value of all harmonic components of reactive power ΣA	(float, lower 2 bytes)	
1088	31088	043F		H		(float, upper 2 bytes)	
1089	31089	0440	QΣA(1)	L	1st harmonic value of reactive power ΣA (fundamental wave)	(float, lower 2 bytes)	
1090	31090	0441		H		(float, upper 2 bytes)	
1091	31091	0442	PhiUi-Uj	L	Phase difference between Ui(1) and Uj(1) of ΣA	(float, lower 2 bytes)	
1092	31092	0443		H		(float, upper 2 bytes)	
1093	31093	0444	PhiUi-Uk	L	Phase difference between Ui(1) and Uk(1) of ΣA	(float, lower 2 bytes)	
1094	31094	0445		H		(float, upper 2 bytes)	
1095	31095	0446	PhiUi-Ii	L	Phase difference between Ui(1) and Ii(1) of ΣA	(float, lower 2 bytes)	
1096	31096	0447		H		(float, upper 2 bytes)	
1097	31097	0448	PhiUj-Ij	L	Phase difference between Uj(1) and Ij(1) of ΣA	(float, lower 2 bytes)	
1098	31098	0449		H		(float, upper 2 bytes)	
1099	31099	044A	PhiUk-Ik	L	Phase difference between Uk(1) and Ik(1) of ΣA	(float, lower 2 bytes)	
1100	31100	044B		H		(float, upper 2 bytes)	

Wiring unit ΣB normal measurement data and delta computation data

1101	31101	044C	UrmsΣB	L	True rms voltage of ΣB	(float, lower 2 bytes)	
1102	31102	044D		H		(float, upper 2 bytes)	
to 1170							

Wiring unit ΣB harmonic measurement data

1171	31171	0492	UΣB(Total)	L	Total value of all harmonic components of voltage ΣB	(float, lower 2 bytes)	
1172	31172	0493		H		(float, upper 2 bytes)	
to 1200							

Wiring unit ΣC normal measurement data and delta computation data

1201	31201	04B0	UrmsΣC	L	True rms voltage of ΣC	(float, lower 2 bytes)	
1202	31202	04B1		H		(float, upper 2 bytes)	
to 1270							

Wiring unit ΣC harmonic measurement data

1271	31271	04F6	UΣC(Total)	L	Total value of all harmonic components of voltage ΣC	(float, lower 2 bytes)	
1272	31272	04F7		H		(float, upper 2 bytes)	
to 1300							

Motor input measurement data

1501	31501	05DC	Speed	L	Rotating speed	(float, lower 2 bytes)	
1502	31502	05DD		H		(float, upper 2 bytes)	
1503	31503	05DE	Torque	L	Torque	(float, lower 2 bytes)	
1504	31504	05DF		H		(float, upper 2 bytes)	
1505	31505	05E0	SyncSp	L	Sync speed	(float, lower 2 bytes)	
1506	31506	05E1		H		(float, upper 2 bytes)	
1507	31507	05E2	Slip	L	Slip	(float, lower 2 bytes)	
1508	31508	05E3		H		(float, upper 2 bytes)	
1509	31509	05E4	Pm	L	Mechanical output of the motor (mechanical power)	(float, lower 2 bytes)	
1510	31510	05E5		H		(float, upper 2 bytes)	

Reg No.	Ref No.	H No.	Register name		Register description		Notes
1511	31511	05E6	SpeedRange	L	Rotating speed range value	(float, lower 2 bytes)	
1512	31512	05E7		H		(float, upper 2 bytes)	
1513	31513	05E8	TorqueRange	L	Torque range value	(float, lower 2 bytes)	
1514	31514	05E9		H		(float, upper 2 bytes)	
1515	31515	05EA	EaU1	L	Electrical angle U1	(float, lower 2 bytes)	
1516	31516	05EB		H		(float, upper 2 bytes)	
1517	31517	05EC	EaI1	L	Electrical angle I1	(float, lower 2 bytes)	
1518	31518	05ED		H		(float, upper 2 bytes)	
1519	31519	05EE	EaU2	L	Electrical angle U2	(float, lower 2 bytes)	
1520	31520	05EF		H		(float, upper 2 bytes)	
1521	31521	05F0	EaI2	L	Electrical angle I2	(float, lower 2 bytes)	
1522	31522	05F1		H		(float, upper 2 bytes)	
1523	31523	05F2	EaU3	L	Electrical angle U3	(float, lower 2 bytes)	
1524	31524	05F3		H		(float, upper 2 bytes)	
1525	31525	05F4	EaI3	L	Electrical angle I3	(float, lower 2 bytes)	
1526	31526	05F5		H		(float, upper 2 bytes)	
1527	31527	05F6	EaU4	L	Electrical angle U4	(float, lower 2 bytes)	
1528	31528	05F7		H		(float, upper 2 bytes)	
1529	31529	05F8	EaI4	L	Electrical angle I4	(float, lower 2 bytes)	
1530	31530	05F9		H		(float, upper 2 bytes)	
1531	31531	05FA	EaU5	L	Electrical angle U5	(float, lower 2 bytes)	
1532	31532	05FB		H		(float, upper 2 bytes)	
1533	31533	05FC	EaI5	L	Electrical angle I5	(float, lower 2 bytes)	
1534	31534	05FD		H		(float, upper 2 bytes)	
1535	31535	05FE	EaU6	L	Electrical angle U6	(float, lower 2 bytes)	
1536	31536	05FF		H		(float, upper 2 bytes)	
1537	31537	0600	EaI6	L	Electrical angle I6	(float, lower 2 bytes)	
1538	31538	0601		H		(float, upper 2 bytes)	

Auxiliary input measurement data

1601	31601	0640	Aux1	L	Auxiliary input 1	(float, lower 2 bytes)	
1602	31602	0641		H		(float, upper 2 bytes)	
1603	31603	0642	Aux2	L	Auxiliary input 2	(float, lower 2 bytes)	
1604	31604	0643		H		(float, upper 2 bytes)	
1605	31605	0644	AuxRange1	L	AUX1 range value	(float, lower 2 bytes)	
1606	31606	0645		H		(float, upper 2 bytes)	
1607	31607	0646	AuxRange2	L	AUX2 range value	(float, lower 2 bytes)	
1608	31608	0647		H		(float, upper 2 bytes)	

Measured data mapped to communication output items (:NUMeric[:NORMal]:ITEM<X> command)

2001+ (X-1)×2			ItemX	L	Measured data mapped to	(float, lower 2 bytes)	Default value
2001+ (X-1)×2+1				H	ItemX	(float, upper 2 bytes)	
2001	32001	07D0	Item1	L	Measured data mapped to	(float, lower 2 bytes)	Urms1
2002	32002	07D1		H	Item 1	(float, upper 2 bytes)	
2003	32003	07D2	Item2	L	Measured data mapped to	(float, lower 2 bytes)	Umn1
2004	32004	07D3		H	Item 2	(float, upper 2 bytes)	
2005	32005	07D4	Item3	L	Measured data mapped to	(float, lower 2 bytes)	Udc1
2006	32006	07D5		H	Item 3	(float, upper 2 bytes)	
2007	32007	07D6	Item4	L	Measured data mapped to	(float, lower 2 bytes)	Uac1
2008	32008	07D7		H	Item 4	(float, upper 2 bytes)	
2009	32009	07D8	Item5	L	Measured data mapped to	(float, lower 2 bytes)	Irms1
2010	32010	07D9		H	Item 5	(float, upper 2 bytes)	
2011	32011	07DA	Item6	L	Measured data mapped to	(float, lower 2 bytes)	Imn1
2012	32012	07DB		H	Item 6	(float, upper 2 bytes)	
2013	32013	07DC	Item7	L	Measured data mapped to	(float, lower 2 bytes)	Idc1
2014	32014	07DD		H	Item 7	(float, upper 2 bytes)	

Reg No.	Ref No.	H No.	Register name		Register description		Notes
2015	32015	07DE	Item8	L	Measured data mapped to	(float, lower 2 bytes)	Iac1
2016	32016	07DF		H	Item 8	(float, upper 2 bytes)	
2017	32017	07E0	Item9	L	Measured data mapped to	(float, lower 2 bytes)	P1
2018	32018	07E1		H	Item 9	(float, upper 2 bytes)	
2019	32019	07E2	Item10	L	Measured data mapped to	(float, lower 2 bytes)	S1
2020	32020	07E3		H	Item 10	(float, upper 2 bytes)	
2021	32021	07E4	Item11	L	Measured data mapped to	(float, lower 2 bytes)	Q1
2022	32022	07E5		H	Item 11	(float, upper 2 bytes)	
2023	32023	07E6	Item12	L	Measured data mapped to	(float, lower 2 bytes)	λ1
2024	32024	07E7		H	Item 12	(float, upper 2 bytes)	
2025	32025	07E8	Item13	L	Measured data mapped to	(float, lower 2 bytes)	φ1
2026	32026	07E9		H	Item 13	(float, upper 2 bytes)	
2027	32027	07EA	Item14	L	Measured data mapped to	(float, lower 2 bytes)	fU1
2028	32028	07EB		H	Item 14	(float, upper 2 bytes)	
2029	32029	07EC	Item15	L	Measured data mapped to	(float, lower 2 bytes)	fI1
2030	32030	07ED		H	Item 15	(float, upper 2 bytes)	
to							
3999	33999	0F9E	Item1000	L	Measured data mapped to	(float, lower 2 bytes)	None
4000	34000	0F9F		H	Item 1000	(float, upper 2 bytes)	

Measured data corresponding to the displayed items of numeric data display (4-value display)

4001+ (X-1)×2			4Items_X	L	Measured data of 4-value display's	(float, lower 2 bytes)	Default value (for the
4001+ (X-1)×2+1				H	item number = X	(float, upper 2 bytes)	6 element model)
4001	34001	0FA0	4Items_1	L	Measured data of 4-value display's	(float, lower 2 bytes)	Urms1
4002	34002	0FA1		H	item number = 1 (page 1)	(float, upper 2 bytes)	
4003	34003	0FA2	4Items_2	L	Measured data of 4-value display's	(float, lower 2 bytes)	Irms1
4004	34004	0FA3		H	item number = 2 (page 1)	(float, upper 2 bytes)	
4005	34005	0FA4	4Items_3	L	Measured data of 4-value display's	(float, lower 2 bytes)	P1
4006	34006	0FA5		H	item number = 3 (page 1)	(float, upper 2 bytes)	
4007	34007	0FA6	4Items_4	L	Measured data of 4-value display's	(float, lower 2 bytes)	λ1
4008	34008	0FA7		H	item number = 4 (page 1)	(float, upper 2 bytes)	
4009	34009	0FA8	4Items_5	L	Measured data of 4-value display's	(float, lower 2 bytes)	Urms2
4010	34010	0FA9		H	item number = 5 (page 2)	(float, upper 2 bytes)	
4011	34011	0FAA	4Items_6	L	Measured data of 4-value display's	(float, lower 2 bytes)	Irms2
4012	34012	0FAB		H	item number = 6 (page 2)	(float, upper 2 bytes)	
4013	34013	0FAC	4Items_7	L	Measured data of 4-value display's	(float, lower 2 bytes)	P2
4014	34014	0FAD		H	item number = 7 (page 2)	(float, upper 2 bytes)	
4015	34015	0FAE	4Items_8	L	Measured data of 4-value display's	(float, lower 2 bytes)	λ2
4016	34016	0FAF		H	item number = 8 (page 2)	(float, upper 2 bytes)	
4017	34017	0FB0	4Items_9	L	Measured data of 4-value display's	(float, lower 2 bytes)	Urms3
4018	34018	0FB1		H	item number = 9 (page 3)	(float, upper 2 bytes)	
to							
4095	34096	0FFE	4Items_48	L	Measured data of 4-value display's	(float, lower 2 bytes)	
4096	34096	0FFF		H	item number = 48 (page 12)	(float, upper 2 bytes)	

Measured data corresponding to the displayed items of numeric data display (8-value display)

4101+ (X-1)×2			8Items_X	L	Measured data of 8-value	(float, lower 2 bytes)	Default value (for the
4101+ (X-1)×2+1				H	display's item number = X	(float, upper 2 bytes)	6 element model)
4101	34101	1004	8Items_1	L	Measured data of 8-value display's	(float, lower 2 bytes)	Urms1
4102	34102	1005		H	item number = 1 (page 1)	(float, upper 2 bytes)	
4103	34103	1006	8Items_2	L	Measured data of 8-value display's	(float, lower 2 bytes)	Irms1
4104	34104	1007		H	item number = 2 (page 1)	(float, upper 2 bytes)	
4105	34105	1008	8Items_3	L	Measured data of 8-value display's	(float, lower 2 bytes)	P1
4106	34106	1009		H	item number = 3 (page 1)	(float, upper 2 bytes)	
4107	34107	100A	8Items_4	L	Measured data of 8-value display's	(float, lower 2 bytes)	S1
4108	34108	100B		H	item number = 4 (page 1)	(float, upper 2 bytes)	

Reg No.	Ref No.	H No.	Register name		Register description		Notes
4109	34109	100C	8Items_5	L	Measured data of 8-value display's	(float, lower 2 bytes)	Q1
4110	34110	100D		H	item number = 5 (page 1)	(float, upper 2 bytes)	
4111	34111	100E	8Items_6	L	Measured data of 8-value display's	(float, lower 2 bytes)	λ1
4112	34112	100F		H	item number = 6 (page 1)	(float, upper 2 bytes)	
4113	34113	1010	8Items_7	L	Measured data of 8-value display's	(float, lower 2 bytes)	φ1
4114	34114	1011		H	item number = 7 (page 1)	(float, upper 2 bytes)	
4115	34115	1012	8Items_8	L	Measured data of 8-value display's	(float, lower 2 bytes)	fU1
4116	34116	1013		H	item number = 8 (page 1)	(float, upper 2 bytes)	
4117	34117	1014	8Items_9	L	Measured data of 8-value display's	(float, lower 2 bytes)	Urms2
4118	34118	1015		H	item number = 9 (page 2)	(float, upper 2 bytes)	
to							
4291	34291	10C2	8Items_96	L	Measured data of 8-value display's	(float, lower 2 bytes)	
4292	34292	10C3		H	item number = 96 (page 12)	(float, upper 2 bytes)	

Measured data corresponding to the displayed items of numeric data display (16-value display)

4301+ (X-1)×2			16Items_X	L	Measured data of 16-value	(float, lower 2 bytes)	Default value (for the
4301+ (X-1)×2+1				H	display's item number = X	(float, upper 2 bytes)	6 element model)
4301	34301	10CC	16Items_1	L	Measured data of 16-value	(float, lower 2 bytes)	Urms1
4302	34302	10CD		H	display's item number = 1 (page 1)	(float, upper 2 bytes)	
4303	34303	10CE	16Items_2	L	Measured data of 16-value	(float, lower 2 bytes)	Irms1
4304	34304	10CF		H	display's item number = 2 (page 1)	(float, upper 2 bytes)	
4305	34305	10D0	16Items_3	L	Measured data of 16-value	(float, lower 2 bytes)	P1
4306	34306	10D1		H	display's item number = 3 (page 1)	(float, upper 2 bytes)	
4307	34307	10D2	16Items_4	L	Measured data of 16-value	(float, lower 2 bytes)	S1
4308	34308	10D3		H	display's item number = 4 (page 1)	(float, upper 2 bytes)	
4309	34309	10D4	16Items_5	L	Measured data of 16-value	(float, lower 2 bytes)	Q1
4310	34310	10D5		H	display's item number = 5 (page 1)	(float, upper 2 bytes)	
4311	34311	10D6	16Items_6	L	Measured data of 16-value	(float, lower 2 bytes)	λ1
4312	34312	10D7		H	display's item number = 6 (page 1)	(float, upper 2 bytes)	
4313	34313	10D8	16Items_7	L	Measured data of 16-value	(float, lower 2 bytes)	φ1
4314	34314	10D9		H	display's item number = 7 (page 1)	(float, upper 2 bytes)	
4315	34315	10DA	16Items_8	L	Measured data of 16-value	(float, lower 2 bytes)	Pc1
4316	34316	10DB		H	display's item number = 8 (page 1)	(float, upper 2 bytes)	
4317	34317	10DC	16Items_9	L	Measured data of 16-value	(float, lower 2 bytes)	fU1
4318	34318	10DD		H	display's item number = 9 (page 1)	(float, upper 2 bytes)	
4319	34319	10DE	16Items_10	L	Measured data of 16-value	(float, lower 2 bytes)	fI1
4320	34320	10DF		H	display's item number = 10 (page 1)	(float, upper 2 bytes)	
4321	34321	10E0	16Items_11	L	Measured data of 16-value	(float, lower 2 bytes)	U+pk1
4322	34322	10E1		H	display's item number = 11 (page 1)	(float, upper 2 bytes)	
4323	34323	10E2	16Items_12	L	Measured data of 16-value	(float, lower 2 bytes)	U-pk1
4324	34324	10E3		H	display's item number = 12 (page 1)	(float, upper 2 bytes)	
4325	34325	10E4	16Items_13	L	Measured data of 16-value	(float, lower 2 bytes)	I+pk1
4326	34326	10E5		H	display's item number = 13 (page 1)	(float, upper 2 bytes)	
4327	34327	10E6	16Items_14	L	Measured data of 16-value	(float, lower 2 bytes)	I-pk1
4328	34328	10E7		H	display's item number = 14 (page 1)	(float, upper 2 bytes)	
4329	34329	10E8	16Items_15	L	Measured data of 16-value	(float, lower 2 bytes)	CfU1
4330	34330	10E9		H	display's item number = 15 (page 1)	(float, upper 2 bytes)	
4331	34331	10EA	16Items_16	L	Measured data of 16-value	(float, lower 2 bytes)	CfI1
4332	34332	10EB		H	display's item number = 16 (page 1)	(float, upper 2 bytes)	
4333	34333	10EC	16Items_17	L	Measured data of 16-value	(float, lower 2 bytes)	Urms2
4334	34334	10ED		H	display's item number = 17 (page 2)	(float, upper 2 bytes)	
to							
4683	34683	124A	16Items_192	L	Measured data of 16-value display's	(float, lower 2 bytes)	
4684	34684	124B		H	item number = 192 (page 12)	(float, upper 2 bytes)	

Reg No.	Ref No.	H No.	Register name		Register description		Notes
Measured data corresponding to the displayed items of numeric data display (Matrix)							
5001+(6×(X-1)+(C-1))×2			Matrix_X_C	L	Measured data of number = X, column = C	(float, lower 2 bytes)	Default value (for the 6 element model)
5001+(6×(X-1)+(C-1))×2+1				H		(float, upper 2 bytes)	
5001	35001	1388	Matrix_1_1	L	Measured data of number = 1, column = 1 (page 1)	(float, lower 2 bytes)	Urms1
5002	35002	1389		H		(float, upper 2 bytes)	
5003	35003	138A	Matrix_1_2	L	Measured data of number = 1, column = 2 (page 1)	(float, lower 2 bytes)	Urms2
5004	35004	138B		H		(float, upper 2 bytes)	
5005	35005	138C	Matrix_1_3	L	Measured data of number = 1, column = 3 (page 1)	(float, lower 2 bytes)	Urms3
5006	35006	138D		H		(float, upper 2 bytes)	
5007	35007	138E	Matrix_1_4	L	Measured data of number = 1, column = 4 (page 1)	(float, lower 2 bytes)	Urms4
5008	35008	138F		H		(float, upper 2 bytes)	
5009	35009	1390	Matrix_1_5	L	Measured data of number = 1, column = 5 (page 1)	(float, lower 2 bytes)	Urms5
5010	35010	1391		H		(float, upper 2 bytes)	
5011	35011	1392	Matrix_1_6	L	Measured data of number = 1, column = 6 (page 1)	(float, lower 2 bytes)	Urms6
5012	35012	1393		H		(float, upper 2 bytes)	
5013	35013	1394	Matrix_2_1	L	Measured data of number = 2, column = 1 (page 1)	(float, lower 2 bytes)	Irms1
5014	35014	1395		H		(float, upper 2 bytes)	
5015	35015	1396	Matrix_2_2	L	Measured data of number = 2, column = 2 (page 1)	(float, lower 2 bytes)	Irms2
5016	35016	1397		H		(float, upper 2 bytes)	
5017	35017	1398	Matrix_2_3	L	Measured data of number = 2, column = 3 (page 1)	(float, lower 2 bytes)	Irms3
5018	35018	1399		H		(float, upper 2 bytes)	
5019	35019	139A	Matrix_2_4	L	Measured data of number = 2, column = 4 (page 1)	(float, lower 2 bytes)	Irms4
5020	35020	139B		H		(float, upper 2 bytes)	
5021	35021	139C	Matrix_2_5	L	Measured data of number = 2, column = 5 (page 1)	(float, lower 2 bytes)	Irms5
5022	35022	139D		H		(float, upper 2 bytes)	
5023	35023	139E	Matrix_2_6	L	Measured data of number = 2, column = 6 (page 1)	(float, lower 2 bytes)	Irms6
5024	35024	139F		H		(float, upper 2 bytes)	
5025	35025	13A0	Matrix_3_1	L	Measured data of number = 3, column = 1 (page 1)	(float, lower 2 bytes)	P1
5026	35026	13A1		H		(float, upper 2 bytes)	
to							
5971	35971	1752	Matrix_81_6	L	Measured data of number = 81, column = 6 (page 9)	(float, lower 2 bytes)	
5972	35972	1753		H		(float, upper 2 bytes)	

Peak over-range status (input register: 0003)

The peak over-range information of each element is mapped to the bits in the following manner.

The bit corresponding to the input in which a peak over-range occurs is set to 1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Aux2	Aux1	Trq	Spd	I6	U6	I5	U5	I4	U4	I3	U3	I2	U2	I1	U1

Float type data

IEEE single-precision floating-point type data. There is no unit prefix.

Data during an error is as follows.

- When the data does not exist (the display shows ”------“)
  NAN 0x7FC00000
- Over-range (the display shows ”--OL-“), computation over-range (the display shows ”--OF-“), error (the display shows ”Error“)
  INF 0x7F800000

Register map (hold register)

Reg No.	Ref No.	H No.	Register name	Register description		Effective range	Default value	BackUp	R/W
Control Data									
0001	40001	0000	Register Hold	Holds and releases register values	(uint 16)	0:release, 1:Hold	0	No	R/W
0002	40002	0001							
0003	40003	0002	INTEG:START/ STOP	Starts and stops integration (all elements)	(uint 16)	0:Stop, 1:Start	0	No	R/W
0004	40004	0003	INTEG:RESET	Resets the integrated value (all elements)	(uint 16)	1:Reset, Not 1: Invalid	-	No	W
0005	40005	0004	INTEG:INDEP	Turns independent element integration on and off	(uint 16)	0:OFF, 1:ON	0	Yes	R/W
0006	40006	0005	INTEG:START	Starts integration* (specified element)	(uint 16)	1 to 63 (specified element)	-	No	W
0007	40007	0006	INTEG:STOP	Stops integration* (specified element)	(uint 16)	1 to 63 (specified element)	-	No	W
0008	40008	0007	INTEG:RESET	Resets the integrated value* (specified element)	(uint 16)	1 to 63 (specified element)	-	No	W
0009	40009	0008	HOLD	Turns on and off the holding of measurement values	(uint 16)	0:OFF, 1:ON	0	Yes	R/W
0010	40010	0009	SINGLE (*TRG)	Executes single measurement	(uint 16)	1: Execute SINGLE, Other than 1: Invalid	-	No	W

* Used when independent integration is on. It is invalid when set to off.

Specifying the target element for integration control (hold registers: 0006, 0007, 0008)

Each element is assigned to the bits as follows.

Set the bit corresponding to the target element for integration control to 1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										EL6	EL5	EL4	EL3	EL2	EL1

8.1 Command Type Compatible with Legacy Models

Many of the functions of this instrument can be controlled using communication commands of legacy models (WT1600, WT1800, WT1800E).
For these functions, you can replace the WT1600, WT1800, or WT1800E with this instrument without having to change the communication programs.

- WT1600: Model 760101
- WT1800: Model WT1801, WT1802, WT1803, WT1804, WT1805, or WT1806
- WT1800E: Model WT1801E, WT1802E, WT1803E, WT1804E, WT1805E, or WT1806E

Remote control settings (Remote Control)

Press **UTILITY**, the **Remote Control** soft key, and then the **Commnad Type** soft key to display the command type choices.

WT-compatible command type

If you specify the WT-compatible command type, the responses and settings of the following communication commands will change.

Response

The response to the following commands will be according to the model of the specified command type.

```
[:INPut]:MODUle?
[:INPut]:POVer?
*IDN?
*OPT?
```

Preset patterns for numeric data output items

The pattern of the numeric data output items preset with the following commands will be according to the model of the specified command type.

:NUMeric[:NORMal]:PRESet
:NUMeric:LIST:PRESet

Default settings for numeric data output items

The numeric data output items of the following commands are initialized as follows when the command type is set.

:NUMeric[:NORMal]:PRESet
:NUMeric:LIST:PRESet
→ **Preset pattern = pattern 2**

:NUMeric[:NORMal]:NUMber → **Number of numeric data output items = 15**
:NUMeric:LIST:NUMber → **Number of numeric list data output items = 1**

Output byte order of numeric data and waveform data (FLOAT format)

When the command type is set, the output byte order (:NUMeric:BYTeorder, :WAVeform:BYTeorder) when the numeric data and waveform data output formats (:NUMeric:FORMat and :WAVeform:FORMat) are set to FLOat is as follows.

Command type setting	WT1800R	Other than WT1800R
:NUMeric:BYTeorder (when :NUMeric:FORMat is FLOat)	LSBFirst	MSBFirst
:WAVeform:BYTeorder (when :WAVeform:FORMat is FLOat)	LSBFirst	MSBFirst

Register output sequence of Modbus/TCP data (float type)

Since one register is 16 bits, float type data is represented with two consecutive registers.
When setting the command type, the arrangement of float type data is as follows:

Command type setting	WT1800R	Other than WT1800R
Arrangement of float type data	Little	Big

* Little = Lower 16 bits followed by upper 16 bits (data order according to the WT1800R specifications)
* Big = Upper 16 bits followed by lower 16 bits (data order according to the WT1800E specifications)

Note

For details on the responses and settings, see the communication interface user's manual for the model corresponding to specified command type.

Appendix 1 Error Messages

This section explains communication error messages.

- On this instrument, the messages can be displayed in the language that you specify with the Menu Language setting on the System Config menu. However, any messages that you read from a PC or other controller will be 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 details on other error messages, see the user's manual, IM WT1801R-02EN.
  - 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 communication errors 300 to 399

  } Listed below

Communication syntax errors (100 to 199)

Code	Message	Corrective Action	Page
102	Syntax error.	A syntax error not covered by error codes 100 to 199.	Chapters 4, 5
103	Invalid separator.	Separate data values with a comma.	4-2
104	Data type error.	Use a correct data type.	Section 4.4
108	Parameter not allowed.	Check the number of data values.	4-8, chapter 5
109	Missing parameter.	Be sure to include all necessary data values.	
111	Header separator error.	Use a space to separate each header from its data.	4-2
112	Program mnemonic too long.	Check the command length.	Chapter 5
113	Undefined header.	Check the header.	
114	Header suffix out of range.	Check the header.	
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, chapter 5
124	Too many digits.	Limit numeric values to 255 digits or less.	
128	Numeric data not allowed.	Use a data type other than <NRf>.	
131	Invalid suffix.	Check the unit of <Voltage>, <Current>, <Time>, or <Frequency>.	4-9
134	Suffix too long.	Check the unit of <Voltage>, <Current>, <Time>, or <Frequency>.	
138	Suffix not allowed.	Units can only be used for <Voltage>, <Current>, <Time>, and <Frequency>.	
141	Invalid character data.	Be sure to select one of the listed choices when the syntax contains {...|...|...}.	Chapters 4, 5
144	Character data too long.	Check the spelling of the strings when the syntax contains {...|...|...}.	Chapter 5
148	Character data not allowed.	Use a data type other than {...|...|...}.	
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.	Chapter 5
158	String data not allowed.	Use a data type other than <string>.	

Code	Message	Corrective Action	Page
161	Invalid block data.	<Block data> cannot be used.	4-10, chapter 5
168	Block data not allowed.	<Block data> cannot be used.	
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.	
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.	
223	Too much data.	Check data byte lengths.	
224	Illegal parameter value.	Check the ranges of the settings.	
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>.	
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.	
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 communication errors (300 and 399)

Code	Message	Corrective Action	Page
300	Communication device-specific error.	Servicing is required. Contact your nearest YOKOGAWA dealer for repairs.	—
399	Fatal error in the communication driver.	Servicing is required. Contact your nearest YOKOGAWA dealer for repairs.	—

Communication warning (50)

Code	Message	Corrective Action	Page
50	*OPC/? exists in message.	Write *OPC or *OPC? at the end of program messages.	—

Other error (350)

Code	Message	Corrective Action	Page
350	Queue overflow.	Read the error queue.	6-8

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 2 History Feature

The history feature records up to the 200 most recent measured numeric data values.

Note

The history feature can only be used on numeric data in normal measurement mode.

Overview of the history feature

- Maximum number of historical entries: 200
  When the maximum number of historical entries is reached, the oldest update data is removed as the latest update data is stored in the history.
- The data in the history is transmitted according to the history communication output mode.

History communication output

When you execute a history communication numeric data query (`:HISTory:NUMeric:VALue?`) or history communication numeric list data query (`:HISTory:NUMeric:LIST:VALue?`), depending on the history communication output mode, various numeric data are output collectively from the history output start position for the number of output counts. For details, see the explanation later in this section.

History communication output mode

You can select the range of history entries to output and the control method of the status register (bit 5: HST).
When the updating of the measured data is completed and the status register (bit 5: HST) changes from 1 to 0, the history communication output start position is set.

N update

When the data is updated the specified number of times (N) after starting a measurement, the status register (bit 5: HST) changes from 1 to 0.

- The range of updated numeric data that is transferred is the specified number of data points (N) before the indication point.
  N: Up to 100

This mode is mainly used to transmit numerous numeric data.
For example, if the update period is short and the data transfer at every measurement update cannot keep up, the communication efficiency can be improved by collectively transferring numeric data over 1 second every second. When the update interval is 50 ms, 1 second of numeric data (20 points) is transferred collectively every second.

For details, see the explanation later in this section.

Explanation

N update mode

In this mode, the completion of updating is indicated every N times the measured data is updated. The update count N is set in advance. The numeric data in the history section indicated automatically is output collectively from the history.

[Operation]

The status register (bit 5: HST) is controlled as follows:

- When the HST status flag (HST) is 1, the status is “Updating.” The status changes to 1 when the reception of the first measured data starts.
- When the HST status flag (HST) is 0, the status is “Not updating.” The status changes to 0 when the updating of the Nth measured data counted from the first measured data is completed.

History communication output count

Manually select the number times to output update data at once during history communication output.

* Cannot be changed during integration.

Selectable range: 1 to 100

This setting is valid when the history communication output mode is N Update.

Set the number of history data points for when performing history communication output continuously with `:HISTory:NUMeric:VALue?` or `:HISTory:NUMeric:LIST:VALue?`. Specify the measurement update count N that is used to indicate “update complete” with the status register (bit 5: HST) when the history communication output mode is N Update.

Status register (Bit 5: HST)

HST is cleared and reset when a measurement is paused. This includes when the settings are changed and when a measurement is held.

[Operation]

- HST is not cleared when a single measurement is executed while the measurement is held. HST is set to “updating” (1) at the updating of the first single measurement and set to “update complete” after the updating of the Nth single measurement. This operation is repeated.
- HST is cleared when a single measurement is executed while the measurement is not held.

History communication output

This feature works only in normal measurement (Normal) mode. If the following queries are made in another mode, “no output” = “#10” (0-byte <block data>) is output.

[Operation]

Numeric data over the reserved history communication output range is secured and output when a history communication output is executed according to the history communication output mode. Therefore, even when there is a history update during a communication output, numeric data over the reserved history communication output range (data over one update period) that has already begun to be transmitted is not overwritten (updated) but is transmitted until the end.

However, if the history communication hold feature is enabled, the numeric data secured at that point is output.

Note

For example, after the range is changed, the history output data remains at “no output” = “#10” (0-byte <block data>) until the first history output data is updated after the measurement is resumed.

History communication numeric data query (:HISTory:NUMeric:VALue?)

- The same item as :NUMeric[:NORMal]:VALue? is output collectively according to the history communication output mode.
- The following commands are shared between :NUMeric[:NORMal]:VALue? and :HISTory:NUMeric:VALue?.

:NUMeric:BYTeorder	Sets or queries the output byte order of the numeric data (FLOAT format).
:NUMeric[:NORMal]:CLEar	Clears the numeric data output items
:NUMeric[:NORMal]:DELete	Deletes the numeric data output items.
:NUMeric[:NORMal]:ITEM<x>	Sets or queries the numeric data output items
:NUMeric[:NORMal]:NUMber	Sets or queries the number of numeric data points to be transmitted
:NUMeric[:NORMal]:PRESet	Presets the numeric data output item pattern.

History communication numeric list data query (:HISTory:NUMeric:LIST:VALue?)

- The same item as :NUMeric:LIST:VALue? is output collectively according to the history communication output mode.
- The following commands are shared between :NUMeric:LIST:VALue? and :HISTory:NUMeric:LIST:VALue?.

:NUMeric:BYTeorder	Sets or queries the output byte order of the numeric data (FLOAT format).
:NUMeric:LIST:CLEar	Clears the harmonic measurement numeric list data output items.
:NUMeric:LIST:DELete	Deletes the harmonic measurement numeric list data output items.
:NUMeric:LIST:ITEM<x>	Sets or queries the output items of the harmonic measurement numeric list data.
:NUMeric:LIST:NUMber	Sets or queries the number of numeric list data points to be transmitted
:NUMeric:LIST:ORDer	Sets or queries the maximum output harmonic order of the harmonic measurement numeric list data.
:NUMeric:LIST:PRESet	Presets the harmonic measurement numeric list data output item pattern.
:NUMeric:LIST:SELect	Sets or queries the output components of the harmonic measurement numeric list data.

History communication hold

Depending on the history communication output mode, history communication output data is secured for the number of counts from the start position at the time when the history communication hold is enabled.

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 features, the subsets that are supported

See section 3.2, "GP-IB Interface Features and 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 UTILITY and then the Remote Ctrl soft key, and changes the address. The new address is valid until the next time it is changed.

(4) Device settings at power-up. The commands that can be used at power-up.

As a basic rule, the previous settings (the settings that were in use when this instrument was turned off) are used.

There are no limitations on the commands that can be used at power-up.

(5) Message exchange options

(a) Input buffer size

1024 bytes

(b) Queries that return multiple response messages

See the examples of the commands given 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

See the examples of the commands given in chapter 5.

(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

Not supported.

(11) Size of the response data block

1 to 2000000

(12) A list of supported common commands

See section 5.23, “Common Command Group” for details.

(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 5.23, “Common Command Group” for details.

(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 5.23, “Common Command Group” for details.

- ***LRN?, *RCL, *SAV**

  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 UTILITY 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 WT1801R-01EN; and the user’s manual, IM WT1801R-02EN.

Appendix 4 Port Numbers

This section describes the port numbers used for communication through this instrument's Ethernet interface and the port numbers that can be used as necessary by this instrument.

Port numbers used by this instrument

Service name	Port number	Communication protocol	Description
VXI-11			See section 1.2
Type			**Connection direction**
PORTMAP (TCP)	111	TCP	PC → This instrument
PORTMAP (UDP)	111	UDP	PC → This instrument
Core channel	11024	TCP	PC → This instrument
Abort channel	11025	TCP	PC → This instrument
Interrupt channel	10002	TCP	This instrument → PC
Modbus/TCP	502	TCP	See section 7.1

Port numbers that can be used by this instrument

The following are commonly used service names and commonly known port numbers for measuring instruments.

Service name	Port number	Communication protocol	Description
FTP-data	20	TCP	File Transfer (Default Data)
FTP	21	TCP	File Transfer (Control)
DNS	53	UDP	Domain Name Server
DHCPv4-server	67	UDP	Bootstrap Protocol Server (DHCP)
DHCPv4-client	68	UDP	Bootstrap Protocol Client (DHCP)
HTTP	80	TCP	World Wide Web HTTP
SNTP	123	UDP	Network Time Protocol
PTP-event	319	UDP	PTP Event
PTP-general	320	UDP	PTP General

On TCP and UDP, port numbers from 1 to 65535 are used to identify the destination application. Multiple port numbers can also be used in a single application. Some port numbers are assigned to specific application services as "commonly known port numbers."

Index

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z